

Your

AN AMIGOS SPECIALIST PUBLICATION

JULY 1984

£1.00

COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

**DETAILED DIRECTORIES
FROM YOUR C64**

**WIN A VIDEO
DIGITISER
WITH NEXUS**

GET ORDERED

DISK FILING FOR YOUR C128

**SPLIT SCREEN
ANTICS**
ON YOUR C16 OR
PLUS/4

C128 ASSEMBLER

GET YOUR NAME ON THE LEADER BOARD

If you have ever bought another golf simulator – shoot yourself!
If you are ever considering it – save yourself a bullet!

ZZAP64

Quite honestly, *Leaderboard* makes all other golf simulations look clumsy and antiquated in comparison.

ZZAP64

This is the sports simulation of the year – if not the decade!

ZZAP64



And swing your way to a record round

Now you can become a golf pro and experience the thrill of having your name up on the "Leaderboard".
With this amazingly realistic simulation you get a true perspective view of your golf game.

As you play on different 18 hole courses on levels varying from amateur to touring professional you'll need skill, concentration and control to come in under par.

CBM 64/128 Coming soon for Spectrum and Amstrad

Features:

- Choice of club, distance, type of shot (hook, slice, puttings) and more
- Realistic Sound
- Practice Driving Range
- Joystick Controlled
- Multiple 18 Hole Golf Courses
- Automatic Scoring



JULY 1988

**VOLUME 2
NUMBER 10**

Editor:
Stuart Cooke

Assistant Editor:
Marie Curry

Advertisement Manager:
John McGarr

**Advertisement Copy
Control:**
Laura Champod

Group Editor:
Dave Bradshaw

**Group Managing
Editor:**
Wendy Palmer

Managing Director:
Peter Neilson

Origination:
Elmly Typesetting

Design:
Argus Design

Editorial & Advertisement Office:
No 1 Gordon Square,
London WC1H 9AB
Telephone: 01-437 0625
Telex: 9871396

Your Commodore magazine
Your 64 is a monthly magazine
appearing on the first Friday of
each month

Distribution by: Argus Press
Sales & Distribution Ltd, 10-18
Pine Street, London EC1A 4JG
Printed by: Alexander Pressons
& Sons Ltd, Tevel, Widdowes,
Kent

Subscription rates: (post
included) to your
Commodore Subscriptions
Department, Inform Ltd, Lincolns
House, 179 The Marrowes,
Luton, Bedfordshire, MK15 9PL

The contents of this publication
including all articles, designs,
plans, drawings and programs
and all copyright and other
intellectual property rights
therein belong to Argus
Specialist Publications Limited.
All rights conferred by the Law
of Copyright and other
intellectual property rights
and by virtue of international
copyright conventions are
specifically reserved to Argus
Specialist Publications Limited
and any reproduction requires
the prior written consent of the
Company. © 1988 Argus
Specialist Publications Limited

FEATURES

■ <i>Cheap Copy</i>	16
Citizen's new printer - for under £50.	
■ <i>Dial-A-File</i>	30
A file editor to complement our Telephone Exchange program.	
■ <i>Remember the Plus/4</i>	36
A look inside the Plus/4's memory.	
■ <i>Graphically Speaking</i>	38
Doodling with CPL's new graphics package.	
■ <i>Amiga Playtime</i>	64
And now there are games for the Amiga.	
■ <i>Game Maker</i>	82
Stuart Cooke reviews Activision's latest utility.	

SERIES

■ <i>Programming the C-16</i>	24
Amass more knowledge about your C16's capabilities.	
■ <i>Froggy</i>	28
Touching up your arcade game.	
■ <i>Programming the 1541</i>	56
Get in a spin with our disk drive series.	
■ <i>Chip Chat</i>	68
Understanding the innermost secrets of your computer.	
■ <i>Going for a Song</i>	70
Musical tricks for your C64.	
■ <i>Welcome to the Machine</i>	84
Teach yourself machine code - made easy.	

REGULARS

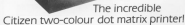
■ <i>Data Statements</i>	5
■ <i>Miscives</i>	10
■ <i>Game of the Month</i>	18
■ <i>Action Replay</i>	45
■ <i>Communication Corner</i>	60
■ <i>Club 128</i>	74
■ <i>Software for Sale</i>	76
■ <i>Ops!</i>	80
■ <i>Listings</i>	96

COMPETITION

■ <i>Detailed Directories</i>	21
A special disk routine for your C64.	
■ <i>The Monster Returns</i>	40
Type in this C64 game.	
■ <i>Assembler 128</i>	62
First part of our Editor Assembler for year 128.	
■ <i>Go Faster 64</i>	68
Speed up your C128 in C64 mode.	
■ <i>Lunar Ordeal</i>	77
A space age games program.	

C
O
N
T
E
N
T
S

COMMODORE COMPATIBLE



- Compatible with Commodore VIC-20, C64, C128, Plus 4, C128 and C1280 Home Computers.
- Helical driver print head — as used on expensive printers.
- Screen-wise dot matrix head.
- Uses 2540/2550 plain paper roll.
- Uses 2540/2550 heavy-duty transparent ribbons.
- Responds to Eijian control codes in dot addressable graphic mode with single and multi-line printing.
- Prints Commodore graphics and inverse characters.
- Programmable line spacing and programmable print capability.

Available from all good
computer stores nationwide.

■ Programmable line spacing and expanded print capability.

Software Shopping

Software is an area of the computer market which is nearly always booming with new products. At the moment there is a vast choice and everyone should find something to please them from utilities to the most basic of mapping games. So let's launch ourselves into the software supermarket straight away.

Something for Everyone

Those of you who have bought and enjoyed New Generation games in the past, will be interested to know that Virgin Games has now acquired all rights to New Gen's back catalogue plus the rights to create new titles.

No one has yet announced which Commodore titles are going to be included in this new deal or given release dates, however Virgin's Nick Alexander was very effusive about the new deal: "I have been a fan of New Generation software since my first days in the business when I saw their 2001 programs at Microfest. We've both come a long way since then and I look forward to further progress."

Chase built, always on the look out for a new way to indulge their reputation as addicts, should look out for Audiogenic's Grand Master Chase. This version has now been successfully available but Audiogenic recently announced that the C-16 game is now totally compatible with the Plus4. £8.95 is the price for C-16, Plus4 and a separate C64 version.

Quickdive has invented a new character - Jerry the punk photographer. He has to wander around snapping new inventions to meet his editor's photography deadline (sounds familiar). Your Commodore's photography boys seem to have an easier life since they haven't got bad mutant monsters trying to nick their film! Look out for Jerry in Quickdive's Neon Force (£19.95) on the C64.

Those who like a bit of recent history intermingled with their gaming can try Microgen's Conflict in Vietnam war game. The game features events dating back to 1954 and features the decisive battles of Ia Drang (1965), Khe Sanh (1968) and Cambella (1970). There is 650 pages of documentation to keep you occupied for hours. In the states it sells for \$39.95 so you'll need to be a fairly dedicated war gamer to afford this one.

Audiogenic is hoping you'll prefer the more leisurely pursuit of a few rounds on the green and try your hand at the new Golf Construction set. Build your own golf course or just play around one of the world famous ones which are reproduced on the program - The Belfry, Wentworth Old Course, Berringle and Royal St George.

DATA

STATEMENTS



An added incentive for those who become really expert is Audiogenic's magnificent offer of £100 worth of golf equipment for the first 12 people who get their handbags down in search (zero - to us estimated) plots. The disk version is on sale for £14.95 and a twin cassette version is priced at £12.95.

Touch Line

Virgin Games: 95-99 Ladbroke Green, London W11 1PG.

Audiogenic: 12 Children Enterprise Centre, Station Rd, Thaxted, Berks RG2 4AA.

Quickdive: Liberty House, 122 Regent St, London W1R 8DB.

Microgen: 120 Lakeside Drive, Hunt Valley, Maryland 21033, USA.

Audiogenic: 60 Long Acre, Covent Garden, London.

Outer Space

AS ALWAYS THERE ARE MASSIVE OF games newly available which have an interstellar setting.

Bubble Fun has come up with a category featuring Blinks - Bio-logically Operated Being (well, aren't we all). The game is Blinks and Blinks has to

scurry through a tunnel and new network inside a planet to collect various bits and pieces to rebuild the planet's core. There are over 500 locations and features including - Anti-grav lifts, teleportal systems, planet surface, security doors, sub planet exploration, propulsion pads and planetary beings (bustle). It's £8.95 on cassette for the C64.

Fans of TV's sick series V will be pleased to know that Genem has released the game of the series on C64 on cassette. For those who missed the telly program, the scenario is involved Blinks and the battles are hard-like aliens whose leader is called Daniel. You play the part of the leader of the resistance movement, Michael Donovan. All you need to do is find out the formula for Red Gen so that you can use it to exterminate the aliens by polluting their air conditioning.

It's available now and costs a mere £5.95.



Immensely arriving in your local computer shop is CRU's new sci-fi adventure, Tan Cent, set on a plague-devastated earth colony in the middle of a far flung interstellar galaxy. Your role is that of the intrepid medical marine who has volunteered to go and repair the damage defence system of the stricken planet so that it can be reconquered. The

only way to do this is shut down the massive fusion reactor which fuels the planet.

If you want to die quickly and often it might be worth a shot at *Exile*.

Touch Line

Bulbrite Bus: 87 High Street, Tonbridge, Kent TN10 1BX.

Cinema 4: Central Street, Manchester M2 5PL.

CMI: CMI House, 9 Kings Yard, Carpenter's Rd, London E15 2AD.

Work, Work, Busy, Busy

Never a dull moment from *ArizolaSoft*. This profit software house has recently released three new utility programs.

The first - *Calcu* (£9.95) - is a tool kit for solving number problems. It should help you sort out the mess which is typically termed your finances. Balance your cheque book, simplify your income tax and develop accurate home and business budgets. In other words it's a very simple to use spreadsheet.

If you use your C64 or C128 to run a business from home then perhaps *ArizolaSoft*'s second utility will be of interest to you. Entitled *BusinessGraph*, it is a visual presentation tool for sales, marketing, forecasting, accounting, management and could also have used for teachers and students. It costs £29.95 on C64/128 disk.

Last, but not least is the new *ArizolaSoft* bunch of utilities is *PaperSparks* with *SpellSparks*. It's a word processing package on C64 disk. *ArizolaSoft* claims that it's time saving and in addition you'll produce totally error free documents because of the 75,000 word dictionary. This one's for the C64 disk and is priced at £29.95.

Back to spreadsheets and *ArizolaSoft*'s successful *Swift* spreadsheet is now available on cassette or disk. The package includes two copies of the program. One runs on the C64 and C128 in 64 mode and the other runs on the 128 and uses the full 128K memory and 80 column display.

ArizolaSoft has also imported *Turbo MBIV* (Multiple Information Retrieval Vehicle) from the states. It loads into the 64 or 128 and runs concurrently with other programs so the user can switch between the main program and the desktop functions of *Turbo MBIV*. Functions include calculator, memo pad, alarm clock, calendar and auto dialler.

Touch Line

ArizolaSoft: 66 Long Ave., Covent Garden, London.

Audiogenic: 12 Children Enterprise Centre, Station Rd, Thame, Berks, RG7 4BA.



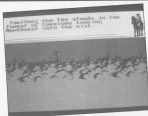
Adventure Spot

IF YOU'RE FED UP WITH DRAB, unexciting adventures then maybe *Adventure House*'s latest offering will bring a breath of fresh air to your life.

The new game is called *Red Hawk* and is billed as a comic strip adventure.

The hero, an ordinary guy called Kevin Oliver alternates between his normal self and Red Hawk. This miraculous transformation is brought about





when he shouts "Kash", super human skills enable him to battle against the villains and criminals in the city. It's available now and costs £8.95.

Beyond has now released its successful Spectrum title - *Decadark's Revenge* - for the C64. It is the sequel to *Dark of Midnight* and contains 5,144 locations and 48,000 views!

It's an adventure cum war game set in a medieval fantasy world. It's a text adventure but there is a difference in that all possible moves can be accomplished by pressing only one key. There's a free audio cassette with the game which tells the story of *Decadark's* with a musical accompaniment. The price is £9.95.

Amstrad has also released a sequel. This one's unimaginatively entitled *Archon II* and - predictably - is the sequel to *Archon*. It's on C64 cassette and costs £9.95.

There's also a new text adventure now out from CRI. The game is called *Plague* and it's for the C64.

It is set in the powerful land of Meridian and you play the role of a young boy who has been given the task to go in search of *The Guardian* - and it's not a newspaper, it's the mysterious protector of the land. The price is £7.95.

Touch Line

Mailroom House: 60 High Street, Hampton Wick, Kingston-upon-Thames, Surrey KT1 4DB.

Beyond: Wellington House, Upper St Martins Lane, London WC2H 9RL.

Amstrad: 47 Long Acree, Covent Garden, London.

CRI: CRI House, 9 King's Yard, Carpenter's Road, London U5 2HD.

Alternative Taste

CRI HAS NOW RELEASED *ROCKE* Harvey Stone on C128.

The company promises that it is the most exciting version of the game to

date using the extra memory available, high definition graphics, new sprites, new animation, new locations, enhanced music and new game play features. It costs £8.95.

No Wave, CRI's alternative software house, has announced the release of *Tubular Bells* for the C64. The program is claimed to be 'entertainment offering amusement for the user in both an active and passive form'. The program has two parts - a sound track and a light synthesiser. The sound track is Mike Oldfield's classic LP generated by computer.

The light synthesiser can be left to run itself or you can intervene to create your own patterns. It should be in the shops now and 750 pennies are required to buy it.

Touch Line

CRI (and No Wave): CRI House, 9 King's Yard, Carpenter's Road, London U5 2HD.

In Touch

COMPUTE B: GETTING INTO THE action with its own multi-user game.

The played title is *Federation II* and there are over 5,000 locations. About 100 of these will be in use at any one time. In comparison, *MUD* has about 400 locations.

Federation II is in the galactic trading genre and Alan Lenton, one of the authors, commented: "Just as *MUD* arose from dungeons and dragons, *Federation II* is a development of the role-playing game. Traveller. It will be very different from *MUD* in concept."

The game is scheduled for an early 1987 launch.

Microsoft has been busy lately, besides its now providing technical support for Microsoft members.

Readers can either send their queries via electronic mail and replies can be found from page 588902.

There is also a hotline where Microsoft members can get their queries

answered immediately. Microsoft members can subscribe to this for £5.

Microsoft has also been involved in another charity exercise.

Capital Radio's Help a London Child appeal has benefited recently to the tune of £100. This was raised from a fund-raised celebrity challenge.

Touch Line

Computers: 7-11 Alameda Road, London NE10 4HJ.

Microsoft: 888, 8 Herby Hill, London EC2R 3EL.

Generally Speaking

FIRST SOFTWARE AND PUBLISHING has decided to kill the computer industry with new launches.

There are 24 new books and nine new software packages scheduled for release this year.

There will also be general reference manuals to examine specific aspects of the C64 and C128. *The Anatomy of the C128* and *Tricks and Tips for the C128* are already available priced at £12.95.

Level 9 Computing is taking a stand against software piracy.

Level 9 has also begun using a Lemlok, a controversial anti-piracy device.

Lemlok tests have been placed at several places in the story of Level 9's game *The Price of Magic*, instead of only at the beginning and the software has been reduced to a third of its original size.

Each Lemlok issued has been printed with the name of the game to avoid confusion.

All Your Commodore readers who consider themselves budding professional programmers can take heart because Superior software has begun a campaign to find some new programmers.

Superior's Richard Hurren said: "This programmer recruitment drive is unique. Only a few companies have used full colour advertising for this purpose before, and, via the adverts, we are offering a free guidebook *Top Tips for Programmers*."

Sales manager Ian Campbell added: "We're looking for programmers of all the major micros (the Spectrum, Commodore, Amstrad, Atari, BBC and Decton)."

So if you want information on this campaign contact Superior or look out for the ads.

Touch Line

First Publishing: Kewthorpe House, 79-81 Margaret Street, London W1N 7HL.

Level 9: PO Box 39, Wenden-Super-Mare, Avon BS24 5UL.

Superior Software: Regent House, 50 Morris Lane, Leeds LS7 1AA.

COMPETITION

**Win a digitiser from Nexus
for your C64.**

THIS MONTH WE'VE SET UP A competition with Nexus which may enable you to win a digitiser for your C64 plus a copy of the Nexus games.

The top prize winner, will be the first person picked out of a hat after the closing date to get the correct solution. The digitiser is worth approximately £150 and there's a copy of the Nexus game thrown in.

There will also be 24 copies of the game as runners up prizes.

The competition is based on the plot of the Nexus game to give you a taste of what you could win.

How to Enter

Study the diagram on this page and then carefully read the following instructions.

1. You are standing in a corridor of the drug-HQ. Your objective is the transmission room from which you can broadcast the facts of your investigation to the world. You are disguised as a

transmission room guard.

2. Exchange places with the transmission room guard, so that YOU are in the transmission room and HE is in the corridor.

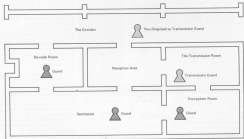
3. There is one key rule - Only one person may occupy a room or the corridor at any time. And only one character - a guard or yourself - may move per turn.

4. How many moves are required to achieve this and what are they?

Instructions

When you have solved the puzzle, fill in the entry form and attach a list of your answers to it on a plain piece of paper. Please write the number of moves you used on the back of your envelope.

Closing date: Friday 25 July 1986.



Nexus Competition

Entry Form

Name

Address

Post code

Number of moves used

Send your entry to: Nexus Competition, Your Correspondence, 1 Golden Square, London W1R 3AB. Closing date: Friday 25 July 1986.

Please attach a sheet describing the moves you used. Please write clearly on the entry form and your answer sheet.

The Rules

Entries will not be accepted from employees of Angus Specialist Publications, Nexus Productions Ltd and Alabaster Peasmore and Sons. This restriction also applies to employees' families and agents of the company.

The How to Enter and Instructions sections form part of the rules. The editor's decision is final and no correspondence will be entered into.

DO YOU WANT TO BE A HERO?



The result of unique co-operation between three ace software developers and the Biggles film production company, Biggles - The Untold Story will knock you right out of the air! It's a multi-part arcade strategy game in which each part must be completed to reach your final goal.

In the air, on the rooftops, on the ground, or in the trenches!

YOU CAN BE A HERO!

Coming in May

Commodore 64
Spectrum 48K

\$9.95 tape \$12.95 disk
\$9.95 tape

Coming in June

Amstrad CPC

\$9.95 tape \$14.95 disk

MIRROSOFT

Rumell Book Centre, Foulton, Bristol BS18 5UG

MISSIVES

Your letters continue to
flood in, keep them coming
so we know what you want.

Decline and Fall

IT CAN BE SAELY SUGGESTED THAT the home computing industry is primarily aimed at those who like computer games and those who wish to develop programming skills and the usage of their machine. However, there appears to be a growing contradiation between these ideas and the content of available publications.

In response to Allen Welsh's letter (April 86, *Your Commodore*), I find it increasingly difficult to find faith with his synopsis that the home computing market is in decline. Virtually all of the British home computing magazines, *Your Commodore* included, offer a disheartening amount of real interest to the average lay bashing fanatic.

As an example, one magazine, although not specifically Commodore oriented, has always been able to offer a cross section of professional programs and utilities for the discerning reader. Unfortunately this format appears to have been re-directed towards more advertising, previews and reviews for hardware and software. Although I understand the need for publication to advertise, and indeed the value of in-depth reviews, I find it difficult to see why this must be achieved at the expense of actual programming (the very basis of home computing), hence a lack of programs, hints, utilities etc. will bring into question the concept of value for money. Subsequent loss of readers will inevitably lead to loss of revenue from advertising until eventually...

Although I have now joined the ICPUK, I have recently found the need to purchase American magazines such as *Compute!* and *EUP* to satisfy my hunger for new ideas and information. Both of these magazines offer good quality diverse programs whilst still advertising. The problems with this situation are that firstly, these magazines are wildly expensive (approx £5) and secondly, in my heat of haste I would much rather purchase an English monthly that can compete, if not surpass, the American competition.

As many magazines have either disappeared or are on the brink of



oblivion, I shall watch future editions of *Your Commodore* with apprehension and perhaps a little hope. As present I am generally pleased with the quality of the magazine and writers such as Allen Welsh continue to perpetuate this quality. *Your Commodore* appears to be the last bastion of sensibility for home computing enthusiasts or are there changes on the way that I should fear?

**A. Lark
Manchester**

am sure you could do this without taking up too much valuable space.

I am interested in purchasing the May cassette providing the program, *Wordbook*, by Ian Murray will be included.

**James O. Butler,
Pikering**

Thank you for your interest James. We have found that our cassettes offer has been immensely popular, but we also appreciate your problem. Firstly, the *Wordbook* program is included on the May cassette. All C64 and C128 programs featured in each issue of *Your Commodore* are included on the cassettes for the relevant month. We are looking at a system whereby we can put a small logo on certain pages to indicate whether the article is one selected for the cassette that month.

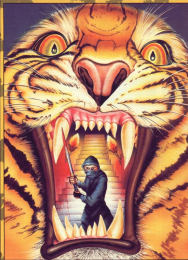
Unfortunately we cannot as yet supply C16 and Plus/4 programs in the same way but we are looking into the possibility of this.

Soft Sale

REGARDING THE SOFTWARE FOR SALE offers in recent *Your Commodore*, might I enquire as to which programs are on the MAY PC/MARBO cassette? Are readers expected to purchase these tapes not knowing exactly which programs they will receive.

Could you indicate at the end of a program or article whether or not it will be on the software for sale cassette. I

THE WAY OF THE TIGER



Under the world of samurai, a large warrior
disappeared into the darkness, leaving a
red blood trail behind him in the night.

Be calm and brave when you find the way
animation and animation control. You'll
find you're looking at the world from a
different point of view.

Experience the stunning effects of the
scrolling action and the master the technique
of the scroll. The scroll is the scroll of the scroll.

And when you find you're surrounded by
enemies, the scroll is the scroll. The scroll is the
scroll of the scroll. The scroll is the scroll of the scroll.

Cassette
£9.95

MSX64K Spectrum 48K Amstrad CBM 64/128

© 1988 by Computer Software Company. All rights reserved. No part of this publication may be reproduced without written permission.

Disc
£14.95



© 1988 by Computer Software Company

D

I

S

K

B

A

S

E

1

2

8

Frank Test helps you to clear up your litter and file it safely.

ONE PROGRAM WHICH THE vast majority of disk drive owners come across, sooner or later, is the massive confusion which can build up when you realize that you have large numbers of disks lying around and you have no idea what's on any of them. When it comes to finding a specific file then the process is annoying and tedious.

Help is at hand, with this program - Disk Base 128. It will move the contents of up to 300 disks on file and help you find what you need.

What It Does

When you run the program, press 'space' at the title page to enter the main program. You will then be faced with 11 icons and a flashing cursor.

To place disk file into the program's memory move the cursor (using the cursor keys - left and shift left) over the Write file icon and press return. You will then be asked for a field number, 1-200. Select your choice and then insert the disk which you want to store in memory in the disk drive. Then, press Return. The program will then load the directory and store it.

When this is complete, press Return to get back to the menu. The contents of that disk are now stored under the field number which you selected.

You may now repeat the process as many times as you

wish, just increment the field number every time.

If you store the same disk twice, at you merely wish to get rid of one which you no longer need, then you can delete it by selecting the Erase file icon. In this case, when asked which field number you wish to erase, simply make your choice and key it in, press Return and the file is then erased.

When you've finished and all your disks are in memory, select the Save file option. You should then enter the number of fields you wish to save, the instance if you only have 40 disks in memory in fields one to 40, then you can just save that number of files. It's quicker and saves memory.

Load files is simply the reverse of save files, select the Load icon, enter the field limit and file name and press Return.

The Scratch icon is so that you can remove an unwanted file from disk without breaking out of the program. Select the Scratch icon, press Return and enter the file name to be erased. It's as easy as that.

The Directory icon will get the directory of a disk and display it on the screen but it will not store in memory, it's just to check and view disks without leaving the program. Select the Dir icon and press Return, the directory will then load, press space to return to the menu.

New disk is not compulsory.

View files icon will display the contents and/or titles of disks in memory. Select View icon and press Return. Then you can select FILES by pressing F or F.

When selecting titles enter the field limit you wish to examine (1-200). The program will then list all the files in memory in that section. Press Return to return.

When selecting Files enter the field number you wish to look at and press space, that will then be displayed to you. Press space to return.

If you know you've got a file somewhere but can't seem to find it select the Find file icon and press Return. You will then be asked what file you're looking for, enter the title and press Return. The program will then search the fields and list any which contain your lost file. Press to return.

Print files will list to the printer the title and contents of a field of your choice. Select the icon, press Return and enter the field you require to be printed.

The icons available to you (from left to right) are: Write file, Erase file, Save file, Load file, Scratch file, Directory, New Disk, View Disk, Find Program, Print file, and file takes you back to the title page, the disks' in memory will not be lost.

The Program

The program is REBAded so you can follow it and the data statements are for ROMs.

Variables

FI - field
dlist() - Dim/array for disk contents
dtp() - Dim array for titles
dtp() - arrays for icon data
PI - lower case
PPI - carriage return


```

396. loc11480232tempPrint"dat
a error:"cmd
398. Fort=Win32API.gdi32.dll
399. mem=
400. closeScreen
402. data255,255,255,128,0,70
,128,0,159,128
404. data1,1,128,1,128,128,0
,240,128,0
406. data270,128,70,100,128,0
,127,128,70,128
408. data270,0,1,100,100,1,0
,112,0,100
410. data270,1,100,1,0,100,10
,1,100,0
412. data1,150,200,0,100,0,0,0
,1,128,0,1
414. data255,255,255,255,0,255
,5,255,128,0,1
416. data258,0,1,128,0,1,150
,240,1,1,25
418. data255,1,125,128,1,128
,63,1,100,0
420. data159,131,63,100,1,100,0
,0,125,131,255,159
422. data129,100,249,170,240
,250,130,180,117,120
424. data258,171,128,17,0,150
,12,17,128,0
426. data607,120,0,0,250,270
,255,0,250,250
428. data255,128,0,1,159,170
,240,150,171,240
430. data249,170,249,150,171,0
,15,159,131,100,159
432. data255,255,159,170,249
,171,128,17,128,17,0
434. data159,131,159,249,170
,240,150,171,240
436. data159,131,159,249,170
,240,150,171,240
438. data159,131,159,249,170
,240,150,171,240
440. data159,131,159,249,170
,240,150,171,240
442. data159,131,159,249,170
,240,150,171,240
444. data159,131,159,249,170
,240,150,171,240
446. data159,131,159,249,170
,240,150,171,240
448. data159,131,159,249,170
,240,150,171,240
450. data159,131,159,249,170
,240,150,171,240
452. data159,131,159,249,170
,240,150,171,240
454. data159,131,159,249,170
,240,150,171,240
456. data159,131,159,249,170
,240,150,171,240
458. data159,131,159,249,170
,240,150,171,240
460. data159,131,159,249,170
,240,150,171,240
462. data159,131,159,249,170
,240,150,171,240
464. data159,131,159,249,170
,240,150,171,240
466. data159,131,159,249,170
,240,150,171,240
468. data159,131,159,249,170
,240,150,171,240
470. data159,131,159,249,170
,240,150,171,240
472. data159,131,159,249,170
,240,150,171,240
474. data159,131,159,249,170
,240,150,171,240
476. data159,131,159,249,170
,240,150,171,240
478. data159,131,159,249,170
,240,150,171,240
480. data159,131,159,249,170
,240,150,171,240
482. data159,131,159,249,170
,240,150,171,240
484. data159,131,159,249,170
,240,150,171,240
486. data159,131,159,249,170
,240,150,171,240
488. data159,131,159,249,170
,240,150,171,240
490. data159,131,159,249,170
,240,150,171,240
492. data159,131,159,249,170
,240,150,171,240
494. data159,131,159,249,170
,240,150,171,240
496. data159,131,159,249,170
,240,150,171,240
498. data159,131,159,249,170
,240,150,171,240
500. data159,131,159,249,170
,240,150,171,240
502. data159,131,159,249,170
,240,150,171,240
504. data159,131,159,249,170
,240,150,171,240
506. data159,131,159,249,170
,240,150,171,240
508. data159,131,159,249,170
,240,150,171,240
510. data159,131,159,249,170
,240,150,171,240
512. data159,131,159,249,170
,240,150,171,240
514. data159,131,159,249,170
,240,150,171,240
516. data159,131,159,249,170
,240,150,171,240
518. data159,131,159,249,170
,240,150,171,240
520. data159,131,159,249,170
,240,150,171,240
522. data159,131,159,249,170
,240,150,171,240
524. data159,131,159,249,170
,240,150,171,240
526. data159,131,159,249,170
,240,150,171,240
528. data159,131,159,249,170
,240,150,171,240
530. data159,131,159,249,170
,240,150,171,240
532. data159,131,159,249,170
,240,150,171,240
534. data159,131,159,249,170
,240,150,171,240
536. data159,131,159,249,170
,240,150,171,240
538. data159,131,159,249,170
,240,150,171,240
540. data159,131,159,249,170
,240,150,171,240
542. data159,131,159,249,170
,240,150,171,240
544. data159,131,159,249,170
,240,150,171,240
546. data159,131,159,249,170
,240,150,171,240
548. data159,131,159,249,170
,240,150,171,240
550. data159,131,159,249,170
,240,150,171,240
552. data159,131,159,249,170
,240,150,171,240
554. data159,131,159,249,170
,240,150,171,240
556. data159,131,159,249,170
,240,150,171,240
558. data159,131,159,249,170
,240,150,171,240
560. data159,131,159,249,170
,240,150,171,240
562. data159,131,159,249,170
,240,150,171,240
564. data159,131,159,249,170
,240,150,171,240
566. data159,131,159,249,170
,240,150,171,240
568. data159,131,159,249,170
,240,150,171,240
570. data159,131,159,249,170
,240,150,171,240
572. data159,131,159,249,170
,240,150,171,240
574. data159,131,159,249,170
,240,150,171,240
576. data159,131,159,249,170
,240,150,171,240
578. data159,131,159,249,170
,240,150,171,240
580. data159,131,159,249,170
,240,150,171,240
582. data159,131,159,249,170
,240,150,171,240
584. data159,131,159,249,170
,240,150,171,240
586. data159,131,159,249,170
,240,150,171,240
588. data159,131,159,249,170
,240,150,171,240
590. data159,131,159,249,170
,240,150,171,240
592. data159,131,159,249,170
,240,150,171,240
594. data159,131,159,249,170
,240
```

THE FINAL CARTRIDGE

THE FIRST OUTSIDE OPERATING
SYSTEM FOR THE CBM 64 *



This new operating system built in a cartridge does not use any memory and is always there. Compatible with 98% of all programs.

DISK TURBO - 8 times faster loading, 5 times faster saving.

SAFE SYSTEM - 10 times faster, even with files - normal Commodore commands - compatible with standard software.

ADVANCED CONTINGENCY INTERFACE - compatible with all the well-known outdoor's printers and Commodore printer programs. Prints all the computer's graphics and control codes (important for testing).

SCREEN DUMP FACILITIES - of various sizes and multipoint (screen). Prints full page with 18 shades of grey for multicolour pictures even from games and programs like Double, Krazy pet, thinking etc. Searches automatically for the memory address of the picture. Special version available for the Commodore 800 and 800+ printers.

THE EXTRA RAM FOR BASIC PROGRAMS AVAILABLE - Easy, new commands: Memory load, Memory write. They move 128 bytes with microprogramming speed anywhere in the 64K Ram of the CBM 64. Can be used with strings and variables.

BASIC 4.0 COMMANDS - like Good, Done, Stopper, Catalog, etc.

BASIC TOOLKIT - with Auto, Remem, find, Goto and Goto2, Find, Help, Get, etc.



Original multicolour full page screen dump print out.

PROGRAMMED FUNCTION KEYS - Run, Load, Save, Catalog, Disk commands, List (removes all list protections).

KEYBOARD EXTRA'S - Allows you to delete part of a line, stop and continue listings, move cursor to lower left-hand corner. Polars and Systeals in files. Ten command operators your printer at a glance.

COMFORTABLE EXTENDED HL-ADAPTER - with extended lead, working up and down. Backspacing, etc. - does not reside in memory.

RESET SWITCH - resets to monitor, resets with add, resets to hi-res printing, resets every protected program.

ON/OFF SWITCH - we hope you never need that one.

FREEZER:

Stops and continues almost every program and allows you to make a total back up to disk or tape automatically.

Speed - Creates one file in disk or tape. Packs the program. Prints 4 to 6 times faster than dedicated freezers.

Many drives:

Freezer systems include: full page printing, fore- and background colour changes, reverse printing, jumps to monitor or reset.

Typing mode:

Auto repeat, collision detection.

12 Months replacement guarantee.

H&P
COMPUTERS

U.K. ORDERS Available to the well-known
Communication Dealers or directly from
H & P Computers
8 Humberstone
Stoke Newington CB6 2JF England 4.
Telephone: 0276 - 85 54 71.

Copyright and trademarked software: H&P Computers.
Registered in U.K. 2002 001. Registered Trademark. No. 000001. Registered Trade Mark. No. 000001.

* works with C128 in the 64 mode

Jayne Golin spends some
time on the fairway and
finds it great fun.

TIGRESS MARKETING ARE VERY PROUD of the latest addition to their ventures. Released through Aristonsoft, the Golf Construction kit should prove to be the ultimate in computer golf simulation until laser-disk games appear.

Whether you're a golfing pro or a rank amateur, this package has everything to recommend it. A full bag of clubs selected by yourself, a range of skills and techniques plus several world famous courses to play on.

Wentworth, Sunningdale, the Belfry and St Georges are all supplied initially and Aristonsoft promises more courses to follow in the future. The construction kit is so detailed that it is possible to recreate the special features of each course from the claustrophobic wooded appearance of Sunningdale to the numerous water hazards of the wet-foot Belfry.

Golfing competitions take two forms: Match Play and Competition.

Competition is the familiar game where the player who holes out on the eighteenth green in the fewest number of strokes is the winner. In this simulation up to four players can compete.

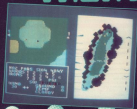
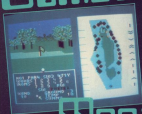
Match Play is a two player game and each hole is either won, lost or halved. This means that the winner is the first person to win 18 holes and the remaining holes are left unplayed. Normally, a drawn match is played on from the first tee until a two hole lead is established by one of the players but this is not possible in this simulation so a drawn match can occur.

As each new hole is played it is loaded from tape to disk. Fortunately this can be described as the normal delay caused by the players walking from green to tee. As the game starts this is hinted at by the legend 'Walking to the first tee' emblazoned across the screen as the rest of the game loads after the initial option screens.

A full complement of clubs cannot be carried so three must be selected for omission at the start of the game. Playing a shot involves several further decisions. The screen design is there to assist your judgement. The top left quadrant of the screen shows the view towards the tee in 3D. Under this is a panel which gives all the vital statistics of the hole and the weather and turf conditions. The right half of the screen shows the plan view of the hole.

Weather and turf conditions affect the length of shot which can be played. Wind direction dictates a particular wobble force on the ball, warm air slows

Game of the Month



Evesham Micros

THE UTILITY SPECIALISTS



Quickdisc+

After extensive evaluation, we have chosen Quickdisc+ as the most reliable, most powerful, and most versatile of the disc-based backup systems.

Just £549 (plus 5% street price) for the Quickdisc+ software (CD-ROM) and the Quickdisc+ backup unit.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

DISC DIRECTOR V4.0

Now the fastest generation of the world's leading file backup utility, Disk Director V4.0 is available. It's the fastest, most powerful, and most versatile of the disc-based backup systems.

Just £549 (plus 5% street price) for the Disk Director V4.0 software (CD-ROM) and the Disk Director V4.0 backup unit.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

THE NEW GENERATION OF BACKUP METHOD: IS NOW EVEN BETTER



THE LAST WORD IN BACKUP TECHNOLOGY

"Freeze Frame" has replaced the old backup method for the past few months. It's the fastest, most powerful, and most versatile of the disc-based backup systems.

Just £549 (plus 5% street price) for the Freeze Frame software (CD-ROM) and the Freeze Frame backup unit.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

ROULETTE

The Fastest Backup Utility

Now the fastest generation of the world's leading file backup utility, Roulette V4.0 is available. It's the fastest, most powerful, and most versatile of the disc-based backup systems.

Just £549 (plus 5% street price) for the Roulette V4.0 software (CD-ROM) and the Roulette V4.0 backup unit.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

Just £549 for 1000 backup copies.

ONLY £39.95

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

ONLY £39.95

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

Free of charge. No hidden costs. No extra charges.

SUBJECT TO AVAILABILITY, ALL ORDERS DESPATCHED THE WORKING DAY AFTER RECEIPT

All orders include P&P and VAT. Send cheque, Postal Order or Credit Card No. Credit Card orders accepted by phone. Callers welcome. Wide range of goods stocked. Trade enquiries welcome. European orders sent price as advertised. Outside Europe £2.00 for airmail. Mail order to Evesham please.

EVESHAM MICROS

BRIDGE STREET, EVESHAM, WORCESTER, WY8 1AH. Tel: 0386 419889

MICRO CENTRE

1750 PERKINS ROAD, COTTRESDALE, BIRMINGHAM. Tel: 021 428 4544

In Australia contact: MICRO ACCESSORIES of S.A., 376 Anderson Road, Smithfield, South Australia 5116. Tel: (08) 254 6204

PROGRAM: DETAILED DIR

```

3 PRINT "CLEAR":POKE 53280,0
  POKE 53281,0
4 IF PEEK(143152)<169 AND PEEK(148152)
  <38 THEN LOAD "V:CODE",0,1
5 CLR:Z=0
6 DATA "DEL ","SEG ","PRG ","USR ","
  "REL ","DEL ","SEG ","PRG ","USR ","
  "REL "
7 FOR J=0 TO 9:READ T(L):NEXT
  L:GOTO 122
8 OPEN 15,0,15,"D":INPUT=15,0,0
  :CLOSE 15
9 IF A<38 THEN PRINT "CLEAR,DOWN",
  RIGHT,WHITE,SPC60:DISK ERROR
  : "[(C)]>A:R4:FOR T=0 TO 255:NEXT
  :RETURN
10 OPEN 4,Z
11 L="":FN4="":L4="":R4="(SPC42)"
12 IF R=1 THEN L4=0
13 OPEN 1,0,0,"R"
14 FOR J=1 TO 14:GET=1,0:NEXT J
15 FOR J=1 TO 16:GET=1,0
  :IF A4=CHR$(168)GOTO 30
16 N4=N4+A4
17 NEXT J:GET=1,0,A4
18 FOR J=1 TO 21:GET=1,0
  :IF A4=CHR$(168)GOTO 35
19 L=L+A4
20 NEXT J
21 FOR J=1 TO 30:GET=1,0:NEXT J
22 PRINT#4,L4:"(C)YR1" : DISK NAME
  "YR1" :ID "114" :0"
23 IF F=8 THEN PRINT#4,L4:"(C)1":
  "TK:SE:FI:BLK: PROGRAM:SPC3:NAME
  :S:AD"
24 IF F=1 THEN PRINT#4,L4:"(I)IDENTA":
  "FILE:BLK: PROGRAM:SPC3:NAME :S:AD
  :L:AD"
25 N4=1:GET=1,K4,T4,S4
  :IF S4="" THEN S4=CHR$(0)
26 F4="" :FOR J=1 TO 16:GET=1,0
  :IF A4=CHR$(168)GOTO 52
27 F4=F4+A4
28 NEXT J
29 FOR J=1 TO 16:GET=1,0:NEXT J
30 L=S4:IF A5<"" THEN L=ASC(A4)
31 GET=1,0,A4:IF R48 THEN GET=1,0,A4
  :GOTO 62
32 R48
33 SW=ST:IF K4=""GOTO 110
34 K=ASC(R4)-128:IF K/4 THEN K=K-50
35 IF K/1 THEN K=0
36 FL=1:IF K/2 OR K/2 THEN FL=0
37 IF F=1 THEN GOTO 78
38 H=MOD(STR$(ASC(T4)),2)
  :PRINT#4,L4:"[(C)]>LEFT4(0%,
  3-LEN(H)):PR4:
39 H=MOD(STR$(ASC(S4)),3)
  :PRINT#4,"WHITE":LEFT4(0%,
  3-LEN(H)):PR4:" "
40 IF F=0 THEN:PRINT#4,"(I)IDENTA":

```

DETAILED DIRECTORIES

Get more details from your
disks with this routine from

L. Jones.

DISK DIRECTORIES ON A C64 CAN only be described as a pain. Unless you have a utility program that will display a directory on your screen the only way of finding what is on your disk is by loading the directory into the computer. Obviously this means the program that was already in there. Hence the need for a program such as this.

The machine code section of this program (DIRECT.COD4) consists of three small routines.

From C60 to C64 is a routine that will display the directory. Once DIRECT.COD4 knows the disk name the command SYS 4952 will display the directory of any disk to the screen, leaving your program intact.

At C64 to C60 is a routine that is used for finding the real address of a program. At C61 to C69 is a small relocatable routine that produces a striped border effect.

Detailed Dir is a program that was written around these three routines to provide a very useful disk utility.

Firstly you can get a listing of any disk on either the screen or printer. If you use the printer then you have the option of printing on the right hand side of the paper. As you are no doubt aware directories only fill up half the width of a standard piece of paper. With this program you can feed the paper back into the printer and use the other half.

Normally the program can also give the following information:

Disk header

Track No and sector No where program is stored.

Number of blocks a program uses.

Program name.

Program start address.

Program end address+1.

If you wish to use any of the machine code routines in your own programs then careful study of Detailed Dir should show you how to use them.

Getting It In

Detailed Dir is a basic program so you should have no problems typing it in.

DIRECT.COD4 is a Basic loader for the machine code. Type this in as a normal program and SAVE it in case you should have made any errors.

Now RUN DIRECT.COD4 and if all is well you should be asked to "PRIN"

```

140 TACK3:
141 IF P=1 THEN PRINT#4,LEFT$(MAGENTA);
LEFT$(B4,13);TACK3:
142 IF P=2 THEN H$=H$(M$(B4(L),2)
:PRINT#4, (GREEN);LEFT$(B4,
3-LEN(H$));H$=" ";
143 IF P=1 THEN H$=H$(B4(L),2)
:PRINT#4, (GREEN);LEFT$(B4,
4-LEN(H$));H$=" ";
144 PRINT#4, (C2);P$=LEFT$(B4,
13-LEN(P$)); (YELLOW);
145 IF K=2 GOTO 128
146 IF K=2 OR K=7 OR P=1 THEN OPEN 3,A,
4, (B4+P$+";"+TACK3)+".R"
147 A$=0:IF FL=1 THEN 22
148 GET#2,A$,B4:A$=0:IF A$=0 THEN B$=0
:CLS:
149 B$=B4 IF B$=0 THEN B$=B$(B4)
150 GOSUB 114
151 IF P=2 THEN 128
152 POKE 255,178:POKE 255,122
:AA=USR(B)
153 IF FL=1 THEN PRINT#4,A;"BYTES":
GOTO 128
154 PRINT#4, (C2); (A$+B$)/255
:AA=A-AA*255:B=B-AA:GOSUB 114
155 CLOSE 2
156 PRINT#4
157 IF S=2 GOTO 45
158 CLOSE 1:CLOSE 4:GOTO 128
159 X=B/16:GOSUB 118:A=X/16
160 FOR J=1 TO 2:AA=X*(X-X/16)*16
:IF AA>0 THEN X=X-AA+2
161 PRINT#4,CHR$(X+48):NEXT J:RETURN
162 RETURN
163 PRINT (CLEAR);A$+ (MAGENTA,
RUSON,SU,RUSOFF):GOSUB 128
164 PRINT (HOME,DOWN,RIGHT,WHITE)
THIS$(SPC,C2);PROGRAM$(SPC,WHITE);
ALLOW YOU TO SPC,C2;PRINT
165 PRINT (DOWN,RIGHT,WHITE)
YOUR DISK DIRECTORY TO THIS$(SPC,C2)
SCREEN(WHITE):PRINT (DOWN,RIGHT,
C2);OR PRINTER.
166 PRINT (DOWN,RIGHT,WHITE)YOU WILL
ALSO HAVE A CHOICE OF THE
167 PRINT (DOWN,RIGHT,WHITE)FOLLOWING
: (C2);END(WHITE,SPC)
OF PROGRAM ADDRESS.
168 PRINT (DOWN,RIGHT);PRINT (C2,SPC)
LEFT(WHITE,SPC);OR (C2,SPC);RIGHT
(WHITE,SPC);OF PAPER.
169 PRINT (DOWN,RIGHT);(C2);CHECK (SPC,
WHITE);YOUR DISK BEFORE YOU CUP
TO
170 PRINT (DOWN,RIGHT);
SCREEN OR PRINTER WITHOUT LOOSING
171 PRINT TAB(13);(DOWN)THIS PROGRAM.
GOSUB 128
172 A$= (GREEN,RUSON,SU,RUSOFF)
:PRINT (CLEAR);GOSUB 128
:PRINT (HOME,DOWN,WHITE,RIGHT,
SPC);FUNCTIONS AT YOUR DISPOSAL
173 PRINT (UP,RED,RIGHT,SPC);(C2);
174 PRINT TAB(13);(C2);(MAGENTA,SPC)
SCREEN DUMP (C2,SPC);
175 PRINT TAB(13);(UP,RIGHT);(RED,C2);
:PRINT (DOWN,RIGHT,YELLOW,SPC);
CHECK (SPC,C2);FOR CORRECT DISK
176 PRINT (DOWN,RIGHT,YELLOW,SPC);
PRINT (SPC,C2);TO SCREEN (WHITE);
(C2);END ADDRESS
177 PRINT (DOWN,RIGHT,YELLOW,SPC);
PRINT (SPC,C2);END ADDRESS
178 PRINT (DOWN,RIGHT,YELLOW,SPC);
NEXT MENU -->>>GOSUB 154:GOTO 128
179 A$= (HOME,DOWN);A$+ (RIGHT);
:AA= (C2,50)---(51)
180 AA= (UP);A$+ (53,50,54)
:AA=AA+AA+AA+AA+AA+AA
181 P$= (55,WHITE,SPC);(C2,SPC,50)
:P$+ (55,WHITE,SPC);(C2,SPC,51)
:P$+ (55,WHITE,SPC);(C2,SPC,52)
:P$+ (55,WHITE,SPC);(C2,SPC,53)
:P$+ (55,WHITE,SPC);(C2,SPC,54)
182 PRINT A$+AA:PRINT A$+AA+P$
:PRINT A$+AA:PRINT A$+AA
:PRINT A$+P$
183 PRINT A$+AA:PRINT A$
:PRINT A$+AA+P$+P$+P$+P$
:PRINT A$+AA:PRINT A$+P$
184 PRINT A$+AA:RETURN
185 SET A$:IF A$= (P$) THEN PRINT
(CLEAR);GOSUB 128:GOTO 128
186 IF A$= (C2) THEN PRINT (CLEAR)
:2=3:GOSUB 254:GOSUB 128:GOTO 148
187 IF A$= (C2) THEN PRINT (CLEAR)
:2=3:GOSUB 255:GOSUB 128:GOTO 148
188 IF A$= (C2) THEN GOTO 228
189 SYS 43377:GOTO 146:REM GOTO128
(C2);43377
190 OPEN 15,B,15, (B4);INPUT#15,A,B4
:CLOSE 15
191 IF A<0 THEN PRINT (DOWN,RIGHT,
WHITE,SPC);DISK ERROR
: (YELLOW);A$=B4:FOR T=8 TO 2555
:NEXT:GOTO 148
192 PRINT (DOWN,RIGHT);(C2,SPC);USE
SPACE BAR TO PAUSE LISTING
193 PRINT TAB(8);(DOWN,RIGHT,C2);ANY
KEY TO RESUME. (FOR DE=1 TO 2555
:NEXT:SYS 43153
194 GOSUB 128:GOTO 148
195 FOR X=1 TO 25:PRINT EN$;NEXT
:FOR X=1 TO 25:PRINT EN$ (DOWN,
LEFT):NEXT
196 FOR X=1 TO 25:PRINT EN$ (LEFT);
:NEXT:FOR X=1 TO 25
:PRINT EN$ (UP,LEFT):NEXT:RETURN
197 PRINT (HOME,DOWN,RIGHT,DOWN,YELLOW)
PRESS ANY KEY TO CONTINUE
198 FOR DE=1 TO 255:NEXT:SYS 43377
199 PRINT (UP,RIGHT);(DOWN,YELLOW)
PRESS ANY KEY TO CONTINUE (RUSOFF)
:FOR DE=1 TO 255:NEXT

```

```

136 GET A$ IF A$="" THEN 138
138 RETURN
282 C$="CYAN,RUSCH,SO,RUSOFF":
  PRINT "CLEAR":GOSUB 188
282 PRINT "HOME,DOWN,WHITE,RIGHT,
  SPC3"FUNCTIONS AT YOUR DISPOSAL"
284 PRINT "UP,RED,RIGHT,SPC8,LT32":
286 PRINT TAB(18):"YELLOW<<<MAGENTA,
  SPC3"PRINTER DUMPS YELLOW,SPC3>>>
288 PRINT TAB(18):"RIGHT4,RED,UP,LT32":
.
218 PRINT "DOWN,RIGHT,YELLOW,SPC3":
  PRINT LEFT$(SPC,CYAN,SPC)
  END ADDRESS"
212 PRINT "DOWN,RIGHT,YELLOW,SPC23":
  PRINT LEFT$(SPC,WHITE,SPC)+(CYAN)
  END ADDRESS"
214 PRINT "DOWN,RIGHT,YELLOW,SPC23":
  PRINT RIGHT$(SPC,WHITE)+CYAN"END
  ADDRESS"
216 PRINT "DOWN,RIGHT,YELLOW,SPC23":
  PRINT RIGHT$(SPC,WHITE)+CYAN"END
  ADDRESS"
218 PRINT "DOWN,RIGHT,YELLOW,SPC23":
  CHANGE DEVICE NUMBER NOW"WHITE,SPC,
  LEFT3":2
220 GOSUB 234:PRINT A$
  IF A$="A3+A4+A5+A6:PRINT A64
  GOTO 222
222 GET A$ IF A$="F1" THEN R$=
  GOSUB 234:GOTO 288
224 IF A$="F3" THEN R$=GOSUB 236
  GOTO 288
226 IF A$="F5" THEN R$=GOSUB 234
  GOTO 288
228 IF A$="F7" THEN R$=GOSUB 236
  GOTO 288
230 IF A$="F9" THEN PRINT "CLEAR":
  GOSUB 188:GOTO 238
232 IF A$="" THEN 2
234 GOSUB 240:GOTO 222
236 PRINT "HOME,DOWN,RIGHT,WHITE,
  SPC3"INPUT A VALUE(SPC,PRESENTA)C
  (GREEN32-255CMAGENTA)3
238 PRINT "DOWN,RIGHT,SPC3"DO NOT
  USE A VALUE OF 3"
240 PRINT "DOWN,RIGHT,SPC4"IS THIS
  IS FOR THE SCREEN"
242 OPEN 1,8:PRINT "LT,DOWN,RIGHT,5,
  SPC3"INPUT DEVICE NUMBER 2 4:SPC2,
  LEFT3":1:INPUT#1,24
244 PRINT "CLOSE 1:2"VAL(24)
  IF 2<8 OR 2>355 OR 2<3 THEN PRINT
  "LTP3":GOTO 236
246 GOSUB 188:GOTO 288
248 PRINT "HOME,DOWN,RIGHT,WHITE,YELLOW":
  PRESS SPACE TO RESTART PROGRAM"UP"
250 FOR DE=1 TO 38:NEXT:SYS 48327
252 PRINT "UP,RIGHT,RUSCH,YELLOW":
  PRESS SPACE TO RESTART PROGRAM"UP3"
  IFOR DE=1 TO 38:NEXT:RETURN
254 F=2:GOSUB 12:RETURN
256 F=1:GOSUB 12:RETURN

```

SPACE TO SAVE. If you have made any errors in the program then correct them before trying again.

When you press space the program "M-CODE" will be saved on to your disk. This is the program that Simulated Dr will load for when you RUN it.

If you want to use the machine code in your own programs then "M-CODE" should be loaded with the extension &1 so that it loads at memory location 49152 (\$C000) onwards.

PROGRAM: DIRECT CODE

```

2888 FOR L=8 TO 15:CHX=RIFOR D=8 TO 15
  READ A(CX+CX+D)POKE 49152+L*16+D,A
NEXT D
2818 READ R:IF A(CX) THEN PRINT "ERROR
  IN LINE":2848+(L+R)STOP
2828 NEXT L
2848 DATA 185,8,133,75,185,8,133,184,
  185,128,133,185,32,68,239,169,1824
2858 DATA 38,133,2,183,1,133,183,133,
  184,185,35,182,8,133,185,134,1851
2868 DATA 185,183,2,182,8,133,187,194,
  188,32,213,243,165,75,32,9,1838
2878 DATA 232,185,185,32,153,232,153,
  8,133,144,168,2,132,183,32,13,2838
2888 DATA 238,133,135,32,15,238,138,
  135,184,144,288,87,184,185,136,295,
  2428
2898 DATA 235,182,8,169,32,32,218,235,
  382,288,258,183,1,141,134,2,2388
2188 DATA 185,185,185,185,32,285,189,
  185,14,141,134,2,169,32,32,2,1863
2118 DATA 231,37,19,238,185,144,288,
  43,281,8,248,8,32,22,231,78,1883
2128 DATA 113,132,32,232,245,248,28,
  32,228,255,248,13,281,32,288,8,2388
2138 DATA 32,228,255,248,254,281,3,
  248,18,189,19,32,22,231,168,2,2883
2148 DATA 78,68,132,32,65,246,32,239,
  232,58,182,8,182,4,149,58,1898
2158 DATA 282,18,251,183,188,133,37,
  182,2,32,198,255,238,181,288,18,
  2328
2168 DATA 238,188,288,8,238,88,288,2,
  238,38,32,228,255,183,144,248,2428
2178 DATA 235,32,284,255,198,37,8,181,
  38,188,38,38,38,38,18,244,1793
2188 DATA 88,182,15,142,32,288,188,73,
  138,288,253,282,234,234,185,182,
  2512
2198 DATA 281,84,248,238,188,248,141,
  32,288,38,8,8,8,238,258,2148
2888 REM ** READY TO SAVE **
2818 PRINT "CLEAR,DOWN,SPC4"DATA
  ALL CORRECT"
2828 PRINT "DOWN,SPC3"PRESS:SPC
  RUSCH:SPC1:RUSOFF:SPC10:SAVE
2838 GET K$ IF K$=" " THEN 2828
2848 POKE 48,8:POKE 44,152:POKE 45,253
  :POKE 46,152:CLR
2858 SAVE "M-CODE",8,1

```

Programming The C-16

IN THIS ARTICLE I SHALL BE concentrating on setting up and using split screens. These are extremely useful in a wide range of games and have the distinct advantage of making available extra memory within the machine while we based on raster interrupts.

Raster Interrupts

The first type of interrupt we discussed last month was the raster interrupt. In order to explain how to use raster interrupts I had better first explain what they are! Rasters basically relate to the screen picture sent by the C-16 to the TV. As the computer outputs the screen signals to the television it scans from the top of the screen to the bottom, 50 times a second. Therefore each pixel line that is generated (there are eight pixel lines in each character line of text) is called a raster line. There are, therefore, 754-288 raster lines for a screen. The border uses a further 171 lines at the top and bottom of the screen display for the PAL colour television system as used in the UK. On the American NTSC system there are only 45 raster lines for the border.

The two memory locations \$F138 and \$F13C (\$E088 and \$E08C decimal) are the vertical raster count registers. These two bytes can be read to find which raster line is currently being displayed. As the picture is displayed from top to bottom, these registers increment from zero to \$11 (or \$B) for the NTSC system, before going back to zero again for the next frame.

Bit zero of address \$F13C (\$E08C decimal) contains the highest bit of the raster compare register and address \$F1D0 (\$E580) contains the lower eight bits. Also, address \$F1F0 (\$E510 decimal) contains the upper eight bits of the nine bit horizontal raster position register. This increments so fast that its only real use to the programmer is to generate random numbers.

Figure 1 shows a short machine code program to

demonstrate the basics behind operating a split screen. The loop in lines \$F000-\$F010 waits for the raster register to equal 120 decimal, i.e. just before halfway down the screen. Lines \$F010-\$F018 then set the colour of the background border to cyan. The loop in lines \$F020-\$F028 waits for the raster scan to equal zero again, i.e. the very top of the screen. Lines \$F029-\$F02F then set the background and border to white — it then goes round again. The result is a two colour screen. Figure 2 shows this coding as code in case you don't have my C-16. Assembler which was published in the June 1985 edition of Your Commodore.

Right, so now we have looked at what rasters are, let's start thinking about raster interrupts.

Addresses \$F10A and \$F10B (\$E290 and \$E291 decimal) contain the nine bits of the raster compare register. Address \$F108 holds the lower eight bits, and bit zero of address \$F10A holds the most significant bit. The remaining bits of \$F10A hold the interrupt mask register, so be careful not to alter any of these when changing zero.

When the raster line count is register \$F13C and \$F138 equals the value in the raster compare register, bit one of the interrupt status register at \$F1F4 is set (see my interrupts article). If bit zero of the interrupt enable register (\$F1F8) is also set, an interrupt is generated. As explained in my previous article addresses \$E074 and

\$E015 (\$B and \$9 decimal) hold the address of the interrupt vector which is usually \$C090, but can be altered to go to a user routine.

Therefore by setting bit one of the interrupt mask register, setting the raster compare registers to the line where you wish to interrupt, and redirecting the interrupt vector you should get an interrupt — right! Well, it's not so simple unfortunately, because the C-16 makes the raster interrupt itself all the time for its own split screen routines for graphics modes two and four. Unfortunately these routines are also active in all the other graphics modes and can mess be turned off.

So why not change the machine's existing split screen routines to create your own interrupts and split screens? Well, you can't, in fact you can't even change the line at which the screen splits. Therefore to create your own raster interrupts, split screens etc., you have to write it all yourself, including some of the interrupt service routines, because these also mess about with the split screen.

So is it all worth the effort? The answer is 'yes'. Split screens are more useful for the C-16 than just about any other computer due to its limited memory capacity. At present, when the high-resolution mode is selected, a huge 10.3 Kbyte portion of the 12.3 Kbyte maximum available memory is used up leaving you with a couple of thousand bytes in which you

can do very little. However if you split the screen and use up 15 of the 25 available high-resolution character lines, leaving the remaining 10 in low-resolution mode, you can save 1280 bytes in the highest screen space plus another 400 bytes in the luminance and 400 bytes in the chrominance tables. We also gain another 600 bytes in the low-res screen and another 600 bytes in the lower colours giving a total of 3200-400-400-600-600-600 bytes saved. Thus we have a total of 7345 bytes free compared with the original 3045. Admittedly this memory is scattered all over the place, but this is not a serious problem for machine code programs. There will be much more about using the additional space in a future article, but now just bear in mind that it is possible to have three and a half times as much memory when in hi-res mode.

Split Screen Routine

Figure 3 shows the assembly listing for the split screen routine. Figure 4 shows the code for the split screen. Enter Figure 4 instead of Figure 3 if you don't have the C-16. Assembler. In Figure 4 lines \$B000-\$B008 load the machine code. The routine is stored in an unused area of memory at \$D008 (1536 decimal) and is around 200 bytes long.

I have included a table which is a breakdown of the system variables used in the split screen program.

Table

Address	Description
1180	position to start the raster line of split 0 (e.g. left split) in the byte 0 to 3
1181	same line number for 1st split
1182	data byte for 1st split
1183	same line number for 2nd split
1184	data byte for 2nd split
1185	same line number for 3rd split
1186	data byte for 3rd split
1187	same line number for 4th split
1188	data byte for 4th split
1189	same line number for 5th split
1190	data byte for 5th split
1191	same line number for 6th split
1192	data byte for 6th split

Note that the position of the splits must follow in order down the screen, i.e. the raster line for the second split must be greater than that for the first split. The raster lines for the screen start at one at the top of the screen and go down to 350 at the bottom. Although the screen has only 350 raster lines and therefore in theory the last line of the screen should be 120+15, for some reason the lines end at 351. Also, if you need an interrupt right at the top of the screen it is better to set the raster line to zero instead of one as the change will take place off the screen avoiding any flicker. To avoid flicker and attribute difficulties, the raster line number for the split should be set from the following equation:

Raster line no. = (Character line) * 1

Note that the character line can be from zero to 24. The equation makes the screen split at the bottom of the character line. Flicker can occur if the raster line is set to the middle of the character line.

Assembly Listing

Lines 10800-10850 — turn on the split screen, altering the position of the interrupt to 0000. They also 'enable' the raster interrupt and set the raster compare register up to the first split.

Lines 10850-10900 — turn off the split screen system by reinitialising the interrupt to the ROM service routine.

Lines 10900-11020 — obtain the interrupt service routine. This routine, mainly adapted from the C-16's ROM, contains all the code

necessary to make it read the keyboard, update the clock, etc. Note that it is not possible in this case to use the C-16's own ROM routines because these would ruin the split screen.

Lines 11020-11050 — perform a split if the interrupt was a raster interrupt.

Lines 11050-11170 — deal with all the RSTZ interrupts used in operating patterns, disk drives, etc.

Lines 11170-11200 — turn off the 'raster interrupt' flag in the raster status register if there was a raster interrupt and also set the new split line position in the raster compare register.

Lines 11200-11300 — decide whether it is a new 1/50th of a second and if it is 0.5, if the next split is the top split, update the clock, read the keyboard and increment raster duration register in the raster at lines 1180-1190. If it is not however, they jump to the IRQ 'wait' ROM routine in line 11300.

Lines 12000-12050 — perform a split. This routine interrupts the Data byte into its appropriate instructions. Because all screen changes must be done at once, it pushes the numbers to be changed into registers 5 and 6 and stores all the values beyond the end of the raster, at lines 12050-12060.

Lines 12050-12110 — break the data byte.

Lines 12110-12180 — deal with the 'TID' fetches from RAM/ROM, i.e. altering bit 2 of address \$FF7E.

Lines 12180-12194 — deal with the 'position of the video matrix' bit by setting byte \$FF70 to 0 for low-res or with the position of the start of the luminance table (stored in address \$0718 for a high-res screen).

Lines 12190-12240 — deal with the '24/25 line screen', the 'bi-map' mode and the 'extended background colour' mode bits by altering bits 1,5 and 6 of register \$FF70 accordingly.

Lines 12240-12290 — deal with the '35/40 columns screen' bit by altering bits 3 and 4 of register \$0707 accordingly.

Lines 12290-12360 — store all these values virtually simultaneously to reduce flicker between splits.

Lines 12360-12390 — move the split position counter on 1 to the next split, or back to the beginning if the end of the split table is reached.

Data Bytes

The data byte for each split contains all the information about the window below it. For example if you want a screen split in the middle with the top half in high-res, and the bottom in low-res, set the data byte for a split at the top of the screen to high-res, and the data byte for the a split half way down for low-res.

Here is how the bits of the data byte are arranged:

Bit 0: 35/40 Columns Screen: 0 for 35 columns and 1 for 40 columns. This is used in smooth scrolling and will be explained in a later article.

Bit 1: Multicolour Mode: 0 for mode 0FF and 1 for mode 0F.

Bit 2: TID Fetches from ROM/RAM. In low-res normal mode and character set is contained in ROM, so TID also runs off device in the C-16; therefore fetches information from ROM. In programmable character mode (see my article in the November 1985 issue) the character set is in RAM, so TID therefore fetches from RAM. In high-res mode also TID fetches from RAM. The bit is 0 for 'TID fetches from RAM' and 1 for 'TID fetches from ROM'.

Bit 3: 24/25 Line Screen. This is also used in smooth scrolling. 24 line mode cuts off half a character line from the top and half a character line from the bottom of the screen, the actual visible effect depends on where the window below the split is. If the window is in the middle of the screen, only, this bit will have no effect at all.

The bit is 0 for 24 lines and 1 for 25 lines.

Bit 4: not used.

Bit 5: Bi-Mapped High-Res mode. The bit is 0 for 0FF (the low-res screen) and 1 for 0F4 (the high-res screen).

Bit 6: Extended Background Colour Mode (see my article in the March 1986 issue of C&E Commodore). The bit is 0 for 0FF and 1 for 0F.

Bit 7: Position of Video Matrix. 0 sets the video matrix address to 2000 for low-res mode. 1 sets the address to the start of the luminance table for high-res mode. Therefore the bit is 0 for low-res and 1 for high-res.

For example to set a normal low-res screen bits 0,2 and 3 should be set, the rest of the bits should be zero. Therefore the data byte would be 1-0-0-0-1-0-0-0 for a standard high-res screen (see multicolour) the byte should be 120-12-0-1-0-0-0-0 = \$45.

Operating Instructions

To turn on the split screen routine type: **SYN 1536.**

To turn off the split screen routine type: **SYN 1568.**

Then to set up a simple split screen with the top half of the screen in high-res and the bottom half in low-res:

POKE 100,4 — for 2 splits = 2.
POKE 100,8 — split at top of the screen

POKE 101,0 — data byte for low-res split

POKE 102,0 — for a split at the 12th line

POKE 103,0 — data byte for 2nd split (low-res)

END 1536 — turn it ON.

The GRAPHIC commands do not have an effect any more, but serve their purpose in specifying which window we are using in Basic. E.g. GRAPHIC 0 specifies low-res, GRAPHIC 1 specifies high-res and GRAPHIC 2 specifies high-res multicolour mode. Also to clear a particular screen put a one after the end of the GRAPHIC command. For instance, if you have entered the above commands in you should now see rubbish on the top half of the screen. Type GRAPHIC 1,1 and the top high-res window will clear. Note that other GRAPHIC 2,1 or GRAPHIC 4,1 clears both the low and the

high-resolution screens simultaneously.

New type CIRCLE,500,45, 45 (don't forget the initial comma to draw a circle in the high-res window).

Then type PAUSE,500,45, to fill the circle in.

Finally type SYS 1588 to get back to normal.

The above example should have given you some ideas on how to use split screens, but does not convey much of the power of these routines. Figure 5 lists a demonstration program which is designed to show up to four screens in action.

Datamake

The following routine, although not having much to do with split screens, is an important utility when dealing with machine code and data. This routine was used when I printed the code in this, and the previous two installments of Programming the C-16. Basically it turns a given block of code into DATA statements, with the option of a checksum at the end of each line. This program greatly reduces the amount of time you spend turning your machine-code games etc. into data so that you can send them to your Commodore!

Figure 6 shows the DATAMAKE program. When the program is RUN enter the START ADDRESS and END ADDRESS of the code in response to the questions. Next enter the starting Basic line number of the block of DATA statements in response to LINE NUMBER 1 and the step between line numbers in response to STEP FOR LINE NUMBERS 2. Lastly enter "Y" or "N" in response to CHECKSUM/Y/N? depending on whether you want a checksum (sum of all the numbers in a line and a useful check) at the end of each line. The program then goes off and constructs the DATA statements. Note that it ends the last line with a "1".

In setting up the DATA statements one or two tricks are used, notably the automatic execution of the lines putting them into memory once every five data statements have been displayed on the screen. This is done by POKEing

characters into the keyboard buffer so that, when the end of line 10158 is reached, commands are stored in the keyboard buffer to make it enter those lines into memory. The actual characters are a ROM-BL directing the cursor to the top of the screen, followed by an INTR instruction. The number of characters in the keyboard buffer is POKE'd into the 'Index to keyboard queue' register at address 239 in line 15100.

The other interesting technique employed in the way the program is initiated automatically is to continue DATA MAKING, every time a new DATA line is POKE'd into memory the computer wipes all the variables, so all the information on cell addresses etc. is lost. To overcome this a line is printed on the screen (for line 15045) that records all the variables in the form of commands, for instance, 5=15000:POKE0, and so on at the end of each group of five DATA lines followed by a GOTO 10018. When the C-16 executes this line all the variables are restored and the program continues execution.

The variable A in line 10080 should be set to the start address of the code. Line 10085 checks the sum of the numbers in each line with the checksum value and if they do not agree the program exits with "DATA ERROR 0Y" followed by the offending line number. This program line should be changed to 10090 L=LNDF and line 10010 should change to

```
10008 POKE 1,A:GOTO 1
```

If the checksum mode was not specified when the data was generated. To put this program at the start of the data and delete the rest of DATAMAKE, delete line 500 then type DELETE 15000-15888.

Well, that concludes this month's article. I hope I've provided some useful routines and some food for thought. Next time I shall be applying this and other routines to creating extra memory on the C-16 and showing how to have 4K for Basic in high-res mode and still have untouched high-res and low-res screens. Impossible! Reserve your copy at the newspaper news

PROGRAM FIGURE 1

```
START 10000 GOTO 10000
10000 GOTO 10000
10010 GOTO 10010
10020 GOTO 10020
10030 GOTO 10030
10040 GOTO 10040
10050 GOTO 10050
10060 GOTO 10060
10070 GOTO 10070
10080 GOTO 10080
10090 GOTO 10090
10100 GOTO 10100
10110 GOTO 10110
10120 GOTO 10120
10130 GOTO 10130
10140 GOTO 10140
10150 GOTO 10150
10160 GOTO 10160
10170 GOTO 10170
10180 GOTO 10180
10190 GOTO 10190
10200 GOTO 10200
10210 GOTO 10210
10220 GOTO 10220
10230 GOTO 10230
10240 GOTO 10240
10250 GOTO 10250
10260 GOTO 10260
10270 GOTO 10270
10280 GOTO 10280
10290 GOTO 10290
10300 GOTO 10300
10310 GOTO 10310
10320 GOTO 10320
10330 GOTO 10330
10340 GOTO 10340
10350 GOTO 10350
10360 GOTO 10360
10370 GOTO 10370
10380 GOTO 10380
10390 GOTO 10390
10400 GOTO 10400
10410 GOTO 10410
10420 GOTO 10420
10430 GOTO 10430
10440 GOTO 10440
10450 GOTO 10450
10460 GOTO 10460
10470 GOTO 10470
10480 GOTO 10480
10490 GOTO 10490
10500 GOTO 10500
10510 GOTO 10510
10520 GOTO 10520
10530 GOTO 10530
10540 GOTO 10540
10550 GOTO 10550
10560 GOTO 10560
10570 GOTO 10570
10580 GOTO 10580
10590 GOTO 10590
10600 GOTO 10600
10610 GOTO 10610
10620 GOTO 10620
10630 GOTO 10630
10640 GOTO 10640
10650 GOTO 10650
10660 GOTO 10660
10670 GOTO 10670
10680 GOTO 10680
10690 GOTO 10690
10700 GOTO 10700
10710 GOTO 10710
10720 GOTO 10720
10730 GOTO 10730
10740 GOTO 10740
10750 GOTO 10750
10760 GOTO 10760
10770 GOTO 10770
10780 GOTO 10780
10790 GOTO 10790
10800 GOTO 10800
10810 GOTO 10810
10820 GOTO 10820
10830 GOTO 10830
10840 GOTO 10840
10850 GOTO 10850
10860 GOTO 10860
10870 GOTO 10870
10880 GOTO 10880
10890 GOTO 10890
10900 GOTO 10900
10910 GOTO 10910
10920 GOTO 10920
10930 GOTO 10930
10940 GOTO 10940
10950 GOTO 10950
10960 GOTO 10960
10970 GOTO 10970
10980 GOTO 10980
10990 GOTO 10990
11000 GOTO 11000
11010 GOTO 11010
11020 GOTO 11020
11030 GOTO 11030
11040 GOTO 11040
11050 GOTO 11050
11060 GOTO 11060
11070 GOTO 11070
11080 GOTO 11080
11090 GOTO 11090
11100 GOTO 11100
11110 GOTO 11110
11120 GOTO 11120
11130 GOTO 11130
11140 GOTO 11140
11150 GOTO 11150
11160 GOTO 11160
11170 GOTO 11170
11180 GOTO 11180
11190 GOTO 11190
11200 GOTO 11200
11210 GOTO 11210
11220 GOTO 11220
11230 GOTO 11230
11240 GOTO 11240
11250 GOTO 11250
11260 GOTO 11260
11270 GOTO 11270
11280 GOTO 11280
11290 GOTO 11290
11300 GOTO 11300
11310 GOTO 11310
11320 GOTO 11320
11330 GOTO 11330
11340 GOTO 11340
11350 GOTO 11350
11360 GOTO 11360
11370 GOTO 11370
11380 GOTO 11380
11390 GOTO 11390
11400 GOTO 11400
11410 GOTO 11410
11420 GOTO 11420
11430 GOTO 11430
11440 GOTO 11440
11450 GOTO 11450
11460 GOTO 11460
11470 GOTO 11470
11480 GOTO 11480
11490 GOTO 11490
11500 GOTO 11500
11510 GOTO 11510
11520 GOTO 11520
11530 GOTO 11530
11540 GOTO 11540
11550 GOTO 11550
11560 GOTO 11560
11570 GOTO 11570
11580 GOTO 11580
11590 GOTO 11590
11600 GOTO 11600
11610 GOTO 11610
11620 GOTO 11620
11630 GOTO 11630
11640 GOTO 11640
11650 GOTO 11650
11660 GOTO 11660
11670 GOTO 11670
11680 GOTO 11680
11690 GOTO 11690
11700 GOTO 11700
11710 GOTO 11710
11720 GOTO 11720
11730 GOTO 11730
11740 GOTO 11740
11750 GOTO 11750
11760 GOTO 11760
11770 GOTO 11770
11780 GOTO 11780
11790 GOTO 11790
11800 GOTO 11800
11810 GOTO 11810
11820 GOTO 11820
11830 GOTO 11830
11840 GOTO 11840
11850 GOTO 11850
11860 GOTO 11860
11870 GOTO 11870
11880 GOTO 11880
11890 GOTO 11890
11900 GOTO 11900
11910 GOTO 11910
11920 GOTO 11920
11930 GOTO 11930
11940 GOTO 11940
11950 GOTO 11950
11960 GOTO 11960
11970 GOTO 11970
11980 GOTO 11980
11990 GOTO 11990
12000 GOTO 12000
12010 GOTO 12010
12020 GOTO 12020
12030 GOTO 12030
12040 GOTO 12040
12050 GOTO 12050
12060 GOTO 12060
12070 GOTO 12070
12080 GOTO 12080
12090 GOTO 12090
12100 GOTO 12100
12110 GOTO 12110
12120 GOTO 12120
12130 GOTO 12130
12140 GOTO 12140
12150 GOTO 12150
12160 GOTO 12160
12170 GOTO 12170
12180 GOTO 12180
12190 GOTO 12190
12200 GOTO 12200
12210 GOTO 12210
12220 GOTO 12220
12230 GOTO 12230
12240 GOTO 12240
12250 GOTO 12250
12260 GOTO 12260
12270 GOTO 12270
12280 GOTO 12280
12290 GOTO 12290
12300 GOTO 12300
12310 GOTO 12310
12320 GOTO 12320
12330 GOTO 12330
12340 GOTO 12340
12350 GOTO 12350
12360 GOTO 12360
12370 GOTO 12370
12380 GOTO 12380
12390 GOTO 12390
12400 GOTO 12400
12410 GOTO 12410
12420 GOTO 12420
12430 GOTO 12430
12440 GOTO 12440
12450 GOTO 12450
12460 GOTO 12460
12470 GOTO 12470
12480 GOTO 12480
12490 GOTO 12490
12500 GOTO 12500
12510 GOTO 12510
12520 GOTO 12520
12530 GOTO 12530
12540 GOTO 12540
12550 GOTO 12550
12560 GOTO 12560
12570 GOTO 12570
12580 GOTO 12580
12590 GOTO 12590
12600 GOTO 12600
12610 GOTO 12610
12620 GOTO 12620
12630 GOTO 12630
12640 GOTO 12640
12650 GOTO 12650
12660 GOTO 12660
12670 GOTO 12670
12680 GOTO 12680
12690 GOTO 12690
12700 GOTO 12700
12710 GOTO 12710
12720 GOTO 12720
12730 GOTO 12730
12740 GOTO 12740
12750 GOTO 12750
12760 GOTO 12760
12770 GOTO 12770
12780 GOTO 12780
12790 GOTO 12790
12800 GOTO 12800
12810 GOTO 12810
12820 GOTO 12820
12830 GOTO 12830
12840 GOTO 12840
12850 GOTO 12850
12860 GOTO 12860
12870 GOTO 12870
12880 GOTO 12880
12890 GOTO 12890
12900 GOTO 12900
12910 GOTO 12910
12920 GOTO 12920
12930 GOTO 12930
12940 GOTO 12940
12950 GOTO 12950
12960 GOTO 12960
12970 GOTO 12970
12980 GOTO 12980
12990 GOTO 12990
13000 GOTO 13000
13010 GOTO 13010
13020 GOTO 13020
13030 GOTO 13030
13040 GOTO 13040
13050 GOTO 13050
13060 GOTO 13060
13070 GOTO 13070
13080 GOTO 13080
13090 GOTO 13090
13100 GOTO 13100
13110 GOTO 13110
13120 GOTO 13120
13130 GOTO 13130
13140 GOTO 13140
13150 GOTO 13150
13160 GOTO 13160
13170 GOTO 13170
13180 GOTO 13180
13190 GOTO 13190
13200 GOTO 13200
13210 GOTO 13210
13220 GOTO 13220
13230 GOTO 13230
13240 GOTO 13240
13250 GOTO 13250
13260 GOTO 13260
13270 GOTO 13270
13280 GOTO 13280
13290 GOTO 13290
13300 GOTO 13300
13310 GOTO 13310
13320 GOTO 13320
13330 GOTO 13330
13340 GOTO 13340
13350 GOTO 13350
13360 GOTO 13360
13370 GOTO 13370
13380 GOTO 13380
13390 GOTO 13390
13400 GOTO 13400
13410 GOTO 13410
13420 GOTO 13420
13430 GOTO 13430
13440 GOTO 13440
13450 GOTO 13450
13460 GOTO 13460
13470 GOTO 13470
13480 GOTO 13480
13490 GOTO 13490
13500 GOTO 13500
13510 GOTO 13510
13520 GOTO 13520
13530 GOTO 13530
13540 GOTO 13540
13550 GOTO 13550
13560 GOTO 13560
13570 GOTO 13570
13580 GOTO 13580
13590 GOTO 13590
13600 GOTO 13600
13610 GOTO 13610
13620 GOTO 13620
13630 GOTO 13630
13640 GOTO 13640
13650 GOTO 13650
13660 GOTO 13660
13670 GOTO 13670
13680 GOTO 13680
13690 GOTO 13690
13700 GOTO 13700
13710 GOTO 13710
13720 GOTO 13720
13730 GOTO 13730
13740 GOTO 13740
13750 GOTO 13750
13760 GOTO 13760
13770 GOTO 13770
13780 GOTO 13780
13790 GOTO 13790
13800 GOTO 13800
13810 GOTO 13810
13820 GOTO 13820
13830 GOTO 13830
13840 GOTO 13840
13850 GOTO 13850
13860 GOTO 13860
13870 GOTO 13870
13880 GOTO 13880
13890 GOTO 13890
13900 GOTO 13900
13910 GOTO 13910
13920 GOTO 13920
13930 GOTO 13930
13940 GOTO 13940
13950 GOTO 13950
13960 GOTO 13960
13970 GOTO 13970
13980 GOTO 13980
13990 GOTO 13990
14000 GOTO 14000
14010 GOTO 14010
14020 GOTO 14020
14030 GOTO 14030
14040 GOTO 14040
14050 GOTO 14050
14060 GOTO 14060
14070 GOTO 14070
14080 GOTO 14080
14090 GOTO 14090
14100 GOTO 14100
14110 GOTO 14110
14120 GOTO 14120
14130 GOTO 14130
14140 GOTO 14140
14150 GOTO 14150
14160 GOTO 14160
14170 GOTO 14170
14180 GOTO 14180
14190 GOTO 14190
14200 GOTO 14200
14210 GOTO 14210
14220 GOTO 14220
14230 GOTO 14230
14240 GOTO 14240
14250 GOTO 14250
14260 GOTO 14260
14270 GOTO 14270
14280 GOTO 14280
14290 GOTO 14290
14300 GOTO 14300
14310 GOTO 14310
14320 GOTO 14320
14330 GOTO 14330
14340 GOTO 14340
14350 GOTO 14350
14360 GOTO 14360
14370 GOTO 14370
14380 GOTO 14380
14390 GOTO 14390
14400 GOTO 14400
14410 GOTO 14410
14420 GOTO 14420
14430 GOTO 14430
14440 GOTO 14440
14450 GOTO 14450
14460 GOTO 14460
14470 GOTO 14470
14480 GOTO 14480
14490 GOTO 14490
14500 GOTO 14500
14510 GOTO 14510
14520 GOTO 14520
14530 GOTO 14530
14540 GOTO 14540
14550 GOTO 14550
14560 GOTO 14560
14570 GOTO 14570
14580 GOTO 14580
14590 GOTO 14590
14600 GOTO 14600
14610 GOTO 14610
14620 GOTO 14620
14630 GOTO 14630
14640 GOTO 14640
14650 GOTO 14650
14660 GOTO 14660
14670 GOTO 14670
14680 GOTO 14680
14690 GOTO 14690
14700 GOTO 14700
14710 GOTO 14710
14720 GOTO 14720
14730 GOTO 14730
14740 GOTO 14740
14750 GOTO 14750
14760 GOTO 14760
14770 GOTO 14770
14780 GOTO 14780
14790 GOTO 14790
14800 GOTO 14800
14810 GOTO 14810
14820 GOTO 14820
14830 GOTO 14830
14840 GOTO 14840
14850 GOTO 14850
14860 GOTO 14860
14870 GOTO 14870
14880 GOTO 14880
14890 GOTO 14890
14900 GOTO 14900
14910 GOTO 14910
14920 GOTO 14920
14930 GOTO 14930
14940 GOTO 14940
14950 GOTO 14950
14960 GOTO 14960
14970 GOTO 14970
14980 GOTO 14980
14990 GOTO 14990
15000 GOTO 15000
15010 GOTO 15010
15020 GOTO 15020
15030 GOTO 15030
15040 GOTO 15040
15050 GOTO 15050
15060 GOTO 15060
15070 GOTO 15070
15080 GOTO 15080
15090 GOTO 15090
15100 GOTO 15100
15110 GOTO 15110
15120 GOTO 15120
15130 GOTO 15130
15140 GOTO 15140
15150 GOTO 15150
15160 GOTO 15160
15170 GOTO 15170
15180 GOTO 15180
15190 GOTO 15190
15200 GOTO 15200
15210 GOTO 15210
15220 GOTO 15220
15230 GOTO 15230
15240 GOTO 15240
15250 GOTO 15250
15260 GOTO 15260
15270 GOTO 15270
15280 GOTO 15280
15290 GOTO 15290
15300 GOTO 15300
15310 GOTO 15310
15320 GOTO 15320
15330 GOTO 15330
15340 GOTO 15340
15350 GOTO 15350
15360 GOTO 15360
15370 GOTO 15370
15380 GOTO 15380
15390 GOTO 15390
15400 GOTO 15400
15410 GOTO 15410
15420 GOTO 15420
15430 GOTO 15430
15440 GOTO 15440
15450 GOTO 15450
15460 GOTO 15460
15470 GOTO 15470
15480 GOTO 15480
15490 GOTO 15490
15500 GOTO 15500
15510 GOTO 15510
15520 GOTO 15520
15530 GOTO 15530
15540 GOTO 15540
15550 GOTO 15550
15560 GOTO 15560
15570 GOTO 15570
15580 GOTO 15580
15590 GOTO 15590
15600 GOTO 15600
15610 GOTO 15610
15620 GOTO 15620
15630 GOTO 15630
15640 GOTO 15640
15650 GOTO 15650
15660 GOTO 15660
15670 GOTO 15670
15680 GOTO 15680
15690 GOTO 15690
15700 GOTO 15700
15710 GOTO 15710
15720 GOTO 15720
15730 GOTO 15730
15740 GOTO 15740
15750 GOTO 15750
15760 GOTO 15760
15770 GOTO 15770
15780 GOTO 15780
15790 GOTO 15790
15800 GOTO 15800
15810 GOTO 15810
15820 GOTO 15820
15830 GOTO 15830
15840 GOTO 15840
15850 GOTO 15850
15860 GOTO 15860
15870 GOTO 15870
15880 GOTO 15880
15890 GOTO 15890
15900 GOTO 15900
15910 GOTO 15910
15920 GOTO 15920
15930 GOTO 15930
15940 GOTO 15940
15950 GOTO 15950
15960 GOTO 15960
15970 GOTO 15970
15980 GOTO 15980
15990 GOTO 15990
16000 GOTO 16000
16010 GOTO 16010
16020 GOTO 16020
16030 GOTO 16030
16040 GOTO 16040
16050 GOTO 16050
16060 GOTO 16060
16070 GOTO 16070
16080 GOTO 16080
16090 GOTO 16090
16100 GOTO 16100
16110 GOTO 16110
16120 GOTO 16120
16130 GOTO 16130
16140 GOTO 16140
16150 GOTO 16150
16160 GOTO 16160
16170 GOTO 16170
16180 GOTO 16180
16190 GOTO 16190
16200 GOTO 16200
16210 GOTO 16210
16220 GOTO 16220
16230 GOTO 16230
16240 GOTO 16240
16250 GOTO 16250
16260 GOTO 16260
16270 GOTO 16270
16280 GOTO 16280
16290 GOTO 16290
16300 GOTO 16300
16310 GOTO 16310
16320 GOTO 16320
16330 GOTO 16330
16340 GOTO 16340
16350 GOTO 16350
16360 GOTO 16360
16370 GOTO 16370
16380 GOTO 16380
16390 GOTO 16390
16400 GOTO 16400
16410 GOTO 16410
16420 GOTO 16420
16430 GOTO 16430
16440 GOTO 16440
16450 GOTO 16450
16460 GOTO 16460
16470 GOTO 16470
16480 GOTO 16480
16490 GOTO 16490
16500 GOTO 16500
16510 GOTO 16510
16520 GOTO 16520
16530 GOTO 16530
16540 GOTO 16540
16550 GOTO 16550
16560 GOTO 16560
16570 GOTO 16570
16580 GOTO 16580
16590 GOTO 16590
16600 GOTO 16600
16610 GOTO 16610
16620 GOTO 16620
16630 GOTO 16630
16640 GOTO 16640
16650 GOTO 16650
16660 GOTO 16660
16670 GOTO 16670
16680 GOTO 16680
16690 GOTO 16690
16700 GOTO 16700
16710 GOTO 16710
16720 GOTO 16720
16730 GOTO 16730
16740 GOTO 16740
16750 GOTO 16750
16760 GOTO 16760
16770 GOTO 16770
16780 GOTO 16780
16790 GOTO 16790
16800 GOTO 16800
16810 GOTO 16810
16820 GOTO 16820
16830 GOTO 16830
16840 GOTO 16840
16850 GOTO 16850
16860 GOTO 16860
16870 GOTO 16870
16880 GOTO 16880
16890 GOTO 16890
16900 GOTO 16900
16910 GOTO 16910
16920 GOTO 16920
16930 GOTO 16930
16940 GOTO 16940
16950 GOTO 16950
16960 GOTO 16960
16970 GOTO 16970
16980 GOTO 16980
16990 GOTO 16990
17000 GOTO 17000
17010 GOTO 17010
17020 GOTO 17020
17030 GOTO 17030
17040 GOTO 17040
17050 GOTO 17050
17060 GOTO 17060
17070 GOTO 17070
17080 GOTO 17080
17090 GOTO 17090
17100 GOTO 17100
17110 GOTO 17110
17120 GOTO 17120
17130 GOTO 17130
17140 GOTO 17140
17150 GOTO 17150
17160 GOTO 17160
17170 GOTO 17170
17180 GOTO 17180
17190 GOTO 17190
17200 GOTO 17200
17210 GOTO 17210
17220 GOTO 17220
17230 GOTO 17230
17240 GOTO 17240
17250 GOTO 17250
17260 GOTO 17260
17270 GOTO 17270
17280 GOTO 17280
17290 GOTO 17290
17300 GOTO 17300
17310 GOTO 17310
17320 GOTO 17320
17330 GOTO 17330
17340 GOTO 17340
17350 GOTO 17350
17360 GOTO 17360
17370 GOTO 17370
17380 GOTO 17380
17390 GOTO 17390
17400 GOTO 17400
17410 GOTO 17410
17420 GOTO 17420
17430 GOTO 17430
17440 GOTO 17440
17450 GOTO 17450
17460 GOTO 17460
17470 GOTO 17470
17480 GOTO 17480
17490 GOTO 17490
17500 GOTO 17500
17510 GOTO 17510
17520 GOTO 17520
17530 GOTO 17530
17540 GOTO 17540
17550 GOTO 17550
17560 GOTO 17560
17570 GOTO 17570
17580 GOTO 17580
17590 GOTO 17590
17600 GOTO 17600
17610 GOTO 17610
17620 GOTO 17620
17630 GOTO 17630
17640 GOTO 17640
17650 GOTO 17650
17660 GOTO 17660
17670 GOTO 17670
17680 GOTO 17680
17690 GOTO 17690
17700 GOTO 17700
17710 GOTO 17710
17720 GOTO 17720
17730 GOTO 17730
17740 GOTO 17740
17750 GOTO 17750
17760 GOTO 17760
```


Daryl Bowers brings
you close to the
completion of your
own arcade game.

FROGGY



THIS MONTH'S INSTALLING us to the point where we have a playable game.

There are two more insertions into the main loop and a new "FINAL" routine.

"DETECT" simply calls the collision detection routines. The hardware sprite collision facility has a serious drawback

—it tells you when one sprite has collided with another, but not which one it has collided with. This is fine for games such as Jet Set Willy, where any sprite collision indicates the death of the player, but in



1070
1080
1090
1100 FINAL
1110
1120
1130
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030
2040
2050
2060
2070
2080
2090
2100
2110
2120
2130
2140
2150
2160
2170
2180
2190
2200
2210
2220
2230
2240
2250
2260
2270
2280
2290
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2400
2410
2420
2430
2440
2450
2460
2470
2480
2490
2500
2510
2520
2530
2540
2550
2560
2570
2580
2590
2600
2610
2620
2630
2640
2650
2660
2670
2680
2690
2700
2710
2720
2730
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950
2960
2970
2980
2990
3000
3010
3020
3030
3040
3050
3060
3070
3080
3090
3100
3110
3120
3130
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
3270
3280
3290
3300
3310
3320
3330
3340
3350
3360
3370
3380
3390
3400
3410
3420
3430
3440
3450
3460
3470
3480
3490
3500
3510
3520
3530
3540
3550
3560
3570
3580
3590
3600
3610
3620
3630
3640
3650
3660
3670
3680
3690
3700
3710
3720
3730
3740
3750
3760
3770
3780
3790
3800
3810
3820
3830
3840
3850
3860
3870
3880
3890
3900
3910
3920
3930
3940
3950
3960
3970
3980
3990
4000
4010
4020
4030
4040
4050
4060
4070
4080
4090
4100
4110
4120
4130
4140
4150
4160
4170
4180
4190
4200
4210
4220
4230
4240
4250
4260
4270
4280
4290
4300
4310
4320
4330
4340
4350
4360
4370
4380
4390
4400
4410
4420
4430
4440
4450
4460
4470
4480
4490
4500
4510
4520
4530
4540
4550
4560
4570
4580
4590
4600
4610
4620
4630
4640
4650
4660
4670
4680
4690
4700
4710
4720
4730
4740
4750
4760
4770
4780
4790
4800
4810
4820
4830
4840
4850
4860
4870
4880
4890
4900
4910
4920
4930
4940
4950
4960
4970
4980
4990
5000
5010
5020
5030
5040
5050
5060
5070
5080
5090
5100
5110
5120
5130
5140
5150
5160
5170
5180
5190
5200
5210
5220
5230
5240
5250
5260
5270
5280
5290
5300
5310
5320
5330
5340
5350
5360
5370
5380
5390
5400
5410
5420
5430
5440
5450
5460
5470
5480
5490
5500
5510
5520
5530
5540
5550
5560
5570
5580
5590
5600
5610
5620
5630
5640
5650
5660
5670
5680
5690
5700
5710
5720
5730
5740
5750
5760
5770
5780
5790
5800
5810
5820
5830
5840
5850
5860
5870
5880
5890
5900
5910
5920
5930
5940
5950
5960
5970
5980
5990
6000
6010
6020
6030
6040
6050
6060
6070
6080
6090
6100
6110
6120
6130
6140
6150
6160
6170
6180
6190
6200
6210
6220
6230
6240
6250
6260
6270
6280
6290
6300
6310
6320
6330
6340
6350
6360
6370
6380
6390
6400
6410
6420
6430
6440
6450
6460
6470
6480
6490
6500
6510
6520
6530
6540
6550
6560
6570
6580
6590
6600
6610
6620
6630
6640
6650
6660
6670
6680
6690
6700
6710
6720
6730
6740
6750
6760
6770
6780
6790
6800
6810
6820
6830
6840
6850
6860
6870
6880
6890
6900
6910
6920
6930
6940
6950
6960
6970
6980
6990
7000
7010
7020
7030
7040
7050
7060
7070
7080
7090
7100
7110
7120
7130
7140
7150
7160
7170
7180
7190
7200
7210
7220
7230
7240
7250
7260
7270
7280
7290
7300
7310
7320
7330
7340
7350
7360
7370
7380
7390
7400
7410
7420
7430
7440
7450
7460
7470
7480
7490
7500
7510
7520
7530
7540
7550
7560
7570
7580
7590
7600
7610
7620
7630
7640
7650
7660
7670
7680
7690
7700
7710
7720
7730
7740
7750
7760
7770
7780
7790
7800
7810
7820
7830
7840
7850
7860
7870
7880
7890
7900
7910
7920
7930
7940
7950
7960
7970
7980
7990
8000
8010
8020
8030
8040
8050
8060
8070
8080
8090
8100
8110
8120
8130
8140
8150
8160
8170
8180
8190
8200
8210
8220
8230
8240
8250
8260
8270
8280
8290
8300
8310
8320
8330
8340
8350
8360
8370
8380
8390
8400
8410
8420
8430
8440
8450
8460
8470
8480
8490
8500
8510
8520
8530
8540
8550
8560
8570
8580
8590
8600
8610
8620
8630
8640
8650
8660
8670
8680
8690
8700
8710
8720
8730
8740
8750
8760
8770
8780
8790
8800
8810
8820
8830
8840
8850
8860
8870
8880
8890
8900
8910
8920
8930
8940
8950
8960
8970
8980
8990
9000
9010
9020
9030
9040
9050
9060
9070
9080
9090
9100
9110
9120
9130
9140
9150
9160
9170
9180
9190
9200
9210
9220
9230
9240
9250
9260
9270
9280
9290
9300
9310
9320
9330
9340
9350
9360
9370
9380
9390
9400
9410
9420
9430
9440
9450
9460
9470
9480
9490
9500
9510
9520
9530
9540
9550
9560
9570
9580
9590
9600
9610
9620
9630
9640
9650
9660
9670
9680
9690
9700
9710
9720
9730
9740
9750
9760
9770
9780
9790
9800
9810
9820
9830
9840
9850
9860
9870
9880
9890
9900
9910
9920
9930
9940
9950
9960
9970
9980
9990
10000
10010
10020
10030
10040
10050
10060
10070
10080
10090
10100
10110
10120
10130
10140
10150
10160
10170
10180
10190
10200
10210
10220
10230
10240
10250
10260
10270
10280
10290
10300
10310
10320
10330
10340
10350
10360
10370
10380
10390
10400
10410
10420
10430
10440
10450
10460
10470
10480
10490
10500
10510
10520
10530
10540
10550
10560
10570
10580
10590
10600
10610
10620
10630
10640
10650
10660
10670
10680
10690
10700
10710
10720
10730
10740
10750
10760
10770
10780
10790
10800
10810
10820
10830
10840
10850
10860
10870
10880
10890
10900
10910
10920
10930
10940
10950
10960
10970
10980
10990
11000
11010
11020
11030
11040
11050
11060
11070
11080
11090
11100
11110
11120
11130
11140
11150
11160
11170
11180
11190
11200
11210
11220
11230
11240
11250
11260
11270
11280
11290
11300
11310
11320
11330
11340
11350
11360
11370
11380
11390
11400
11410
11420
11430
11440
11450
11460
11470
11480
11490
11500
11510
11520
11530
11540
11550
11560
11570
11580
11590
11600
11610
11620
11630
11640
11650
11660
11670
11680
11690
11700
11710
11720
11730
11740
11750
11760
11770
11780
11790
11800
11810
11820
11830
11840
11850
11860
11870
11880
11890
11900
11910
11920
11930
11940
11950
11960
11970
11980
11990
12000
12010
12020
12030
12040
12050
12060
12070
12080
12090
12100
12110
12120
12130
12140
12150
12160
12170
12180
12190
12200
12210
12220
12230
12240
12250
12260
12270
12280
12290
12300
12310
12320
12330
12340
12350
12360
12370
12380
12390
12400
12410
12420
12430
12440
12450
12460
12470
12480
12490
12500
12510
12520
12530
12540
12550
12560
12570
12580
12590
12600
12610
12620
12630
12640
12650
12660
12670
12680
12690
12700
12710
12720
12730
12740
12750
12760
12770
12780
12790
12800
12810
12820
12830
12840
12850
12860
12870
12880
12890
12900
12910
12920
12930
12940
12950
12960
12970
12980
12990
13000
13010
13020
13030
13040
13050
13060
13070
13080
13090
13100
13110
13120
13130
13140
13150
13160
13170
13180
13190
13200
13210
13220
13230
13240
13250
13260
13270
13280
13290
13300
13310
13320
13330
13340
13350
13360
13370
13380
13390
13400
13410
13420
13430
13440
13450
13460
13470
13480
13490
13500
13510
13520
13530
13540
13550
13560
13570
13580
13590
13600
13610
13620
13630
13640
13650
13660
13670
13680
13690
13700
13710
13720
13730
13740
13750
13760
13770
13780
13790
13800
13810
13820
13830
13840
13850
13860
13870
13880
13890
13900
13910
13920
13930
13940
13950
13960
13970
13980
13990
14000
14010
14020
14030
14040
14050
14060
14070
14080
14090
14100
14110
14120
14130
14140
14150
14160
14170
14180
14190
14200
14210
14220
14230
14240
14250
14260
14270
14280
14290
14300
14310
14320
14330
14340
14350
14360
14370
14380
14390
14400
14410
14420
14430
14440
14450
14460
14470
14480
14490
14500
14510
14520
14530
14540
14550
14560
14570
14580
14590
14600
14610
14620
14630
14640
14650
14660
14670
14680
14690
14700
14710
14720
14730
14740
14750
14760
14770
14780
14790
14800
14810
14820
14830
14840
14850
14860
14870
14880
14890
14900
14910
14920
14930
14940
14950
14960
14970
14980
14990
15000
15010
15020
15030
15040
15050
15060
15070
15080
15090
15100
15110
15120
15130
15140
15150
15160
15170
15180
15190
15200
15210
15220
15230
15240
15250
15260
15270
15280
15290
15300
15310
15320
15330
15340
15350
15360
15370
15380
15390
15400
15410
15420
15430
15440
15450
15460
15470
15480
15490
15500
15510
15520
15530
15540
15550
15560
15570
15580
15590
15600
15610
15620
15630
15640
15650
15660
15670
15680
15690
15700
15710
15720
15730
15740
15750
15760
15770
15780
15790
15800
15810
15820
15830
15840
15850
15860
15870
15880
15890
15900
15910
15920
15930
15940
15950
15960
15970
15980
15990
16000
16010
16020
16030
16040
16050
16060
16070
16080
16090
16100
16110
16120
16130
16140
16150
16160
16170
16180
16190
16200
16210
16220
16230
16240
16250
16260
16270
16280
16290
16300
16310
16320
16330
16340
16350
16360
16370
16380
16390
16400
16410
16420
16430
16440
16450
16460
16470
16480
16490
16500
16510
16520
16530
16540
16550
16560
16570
16580
16590
16600
16610
16620
16630
16640
16650
16660
16670
16680
16690
16700
16710
16720
16730
16740
16750
16760
16770
16780
16790
16800
16810
16820
16830
16840
16850
16860
16870
16880
16890
16900
16910
16920
16930
16940
16950
16960
16970
16980
16990
17000
17010
17020
17030
17040
17050
17060
17070
17080
17090
17100
17110
17120
17130
17140
17150
17160
17170
17180
17190
17200
17210
17220
17230
17240
17250
17260
17270
17280
17290
17300
17310
17320
17330
17340
17350
17360
17370
17380
17390
17400
17410
17420
17430
17440
17450
17460
17470
17480
17490
17500
17510
17520
17530
17540
17550
17560
17570
17580
17590
17600
17610
17620
17630
17640
17650
17660
17670
17680
17690
17700
17710
17720
17730
17740
17750
17760
17770
17780
17790
17800
17810
17820
17830
17840
17850
17860
17870
17880
17890
17900
17910
17920
17930
17940
17950
17960
17970
17980
17990
18000
18010
18020
18030
18040
18050
18060
18070
18080
18090
18100
18110
18120
18130
18140
18150
18160
18170
18180
18190
18200
18210
18220
18230
18240
18250
18260
18270
18280
18290
18300
18310
18320
18330
18340
18350
18360
18370
18380
18390
18400
18410
18420
18430
18440
18450
18460
18470
18480
18490
18500
18510
18520
18530
18540
18550
18560
18570
18580
18590
18600
18610
18620
18630
18640
18650
18660
18670
18680
18690
18700
18710
18720
18730
18740
18750
18760
18770
18780
18790
18800
18810
18820
18830
18840
18850
18860
18870
18880
18890
18900
18910
18920
18930
18940
18950
18960
18970
18980
18990
19000
19010
19020
19030
19040
19050
19060
19070
19080
19090
19100
19110
19120
19130
19140
19150
19160
19170
19180
19190
19200
19210
19220
19230
19240
19250
19260
19270
19280
19290
19300
19310
19320
19330
19

mind games, this will not suffice. For this reason I have substituted collisions by comparing the X and Y coordinates of the frog and the other characters.

RELOCATE is the first routine to be called. This checks if the frog is sitting in a pothole. The route from the start to "CRUIT" is used to ascertain that the frog is sitting down. To check this the current sprite definition is found — if this is equal to 300 then the coordinates are checked.

To see whether the characters underneath him are potholes, we read first the position of the relevant characters. Remember that the first middle quote X coordinate is 31 and that the frog graphics start 17 pixels into the front sprite so must subtract 48 from the X position. If we divide this value by eight (the width of any character in pixels) then we have the X character position of the frog.

"LOOPIT" checks to see if the next seven characters are

potholes or not, and if they are we go to "CRUIT". At this point we decrease the delay "RECHISEL" to reduce the damage sustained for each contact, then if this has reached zero we reduce the "TICKS" left.

"RECHISEL" is a very simple check. We take the Frenchman's position, add 8, and store this in BAC. If the frog's X position is less than this, he is dead.

"PLUCK" follows the same comparisons, with the addition that the frog must

be jumping, and if a collision takes place, the fly's position is reset to the start again.

"WASH" transfers the variable block to the end of the program in order that they can be retrieved at the start of a new game. "WASH" does exactly the opposite, and is called in "FIND" to reset all variables to their original values.

The final routine is "RECHISEL" which operates in the same way as "PLUCK".

Next month — *Arks*.

```

13140      I
13150      LDA FLYX01
13160      BNE HITTEST
13170      I
13180      LDA STAB01,Y
13190      CLC
13200      ADC #0
13210      CMP FLYX01
13220      BCC NOTFLY
13230      CLC
13240      ADC #0
13250      BCC HITTEST
13260      CMP FLYX01
13270      BCC NOTFLY
13280      I
13290      LDA #0
13300      STA FLYX01
13310      LDA #0
13320      STA FLYX01
13330      LDA #0
13340      STA FLYX01
13350      LDA #0
13360      STA FLYX01
13370      LDA #0
13380      STA FLYX01
13390      LDA #0
13400      STA FLYX01
13410      LDA #0
13420      STA FLYX01
13430      LDA #0
13440      STA FLYX01
13450      LDA #0
13460      STA FLYX01
13470      LDA #0
13480      STA FLYX01
13490      LDA #0
13500      STA FLYX01
13510      LDA #0
13520      STA FLYX01
13530      LDA #0
13540      STA FLYX01
13550      LDA #0
13560      STA FLYX01
13570      LDA #0
13580      STA FLYX01
13590      LDA #0
13600      STA FLYX01
13610      LDA #0
13620      STA FLYX01
13630      LDA #0
13640      STA FLYX01
13650      LDA #0
13660      STA FLYX01
13670      LDA #0
13680      STA FLYX01
13690      LDA #0
13700      STA FLYX01
13710      LDA #0
13720      STA FLYX01
13730      LDA #0
13740      STA FLYX01
13750      LDA #0
13760      STA FLYX01
13770      LDA #0
13780      STA FLYX01
13790      LDA #0
13800      STA FLYX01
13810      LDA #0
13820      STA FLYX01
13830      LDA #0
13840      STA FLYX01
13850      LDA #0
13860      STA FLYX01
13870      LDA #0
13880      STA FLYX01
13890      LDA #0
13900      STA FLYX01
13910      LDA #0
13920      STA FLYX01
13930      LDA #0
13940      STA FLYX01
13950      LDA #0
13960      STA FLYX01
13970      LDA #0
13980      STA FLYX01
13990      LDA #0
14000      STA FLYX01

```

```

13400      I
13410      LDA FLYX01
13420      BNE HITTEST
13430      I
13440      LDA STAB01,Y
13450      CLC
13460      ADC #0
13470      CMP FLYX01
13480      BCC NOTFLY
13490      CLC
13500      ADC #0
13510      BCC HITTEST
13520      CMP FLYX01
13530      BCC NOTFLY
13540      I
13550      LDA #0
13560      STA FLYX01
13570      LDA #0
13580      STA FLYX01
13590      LDA #0
13600      STA FLYX01
13610      LDA #0
13620      STA FLYX01
13630      LDA #0
13640      STA FLYX01
13650      LDA #0
13660      STA FLYX01
13670      LDA #0
13680      STA FLYX01
13690      LDA #0
13700      STA FLYX01
13710      LDA #0
13720      STA FLYX01
13730      LDA #0
13740      STA FLYX01
13750      LDA #0
13760      STA FLYX01
13770      LDA #0
13780      STA FLYX01
13790      LDA #0
13800      STA FLYX01
13810      LDA #0
13820      STA FLYX01
13830      LDA #0
13840      STA FLYX01
13850      LDA #0
13860      STA FLYX01
13870      LDA #0
13880      STA FLYX01
13890      LDA #0
13900      STA FLYX01
13910      LDA #0
13920      STA FLYX01
13930      LDA #0
13940      STA FLYX01
13950      LDA #0
13960      STA FLYX01
13970      LDA #0
13980      STA FLYX01
13990      LDA #0
14000      STA FLYX01

```

```

14000      BNECHISEL
14010      BNECHISEL
14020      BNECHISEL
14030      BNECHISEL
14040      BNECHISEL
14050      BNECHISEL
14060      BNECHISEL
14070      BNECHISEL
14080      BNECHISEL
14090      BNECHISEL
14100      BNECHISEL
14110      BNECHISEL
14120      BNECHISEL
14130      BNECHISEL
14140      BNECHISEL
14150      BNECHISEL
14160      BNECHISEL
14170      BNECHISEL
14180      BNECHISEL
14190      BNECHISEL
14200      BNECHISEL
14210      BNECHISEL
14220      BNECHISEL
14230      BNECHISEL
14240      BNECHISEL
14250      BNECHISEL
14260      BNECHISEL
14270      BNECHISEL
14280      BNECHISEL
14290      BNECHISEL
14300      BNECHISEL
14310      BNECHISEL
14320      BNECHISEL
14330      BNECHISEL
14340      BNECHISEL
14350      BNECHISEL
14360      BNECHISEL
14370      BNECHISEL
14380      BNECHISEL
14390      BNECHISEL
14400      BNECHISEL
14410      BNECHISEL
14420      BNECHISEL
14430      BNECHISEL
14440      BNECHISEL
14450      BNECHISEL
14460      BNECHISEL
14470      BNECHISEL
14480      BNECHISEL
14490      BNECHISEL
14500      BNECHISEL
14510      BNECHISEL
14520      BNECHISEL
14530      BNECHISEL
14540      BNECHISEL
14550      BNECHISEL
14560      BNECHISEL
14570      BNECHISEL
14580      BNECHISEL
14590      BNECHISEL
14600      BNECHISEL
14610      BNECHISEL
14620      BNECHISEL
14630      BNECHISEL
14640      BNECHISEL
14650      BNECHISEL
14660      BNECHISEL
14670      BNECHISEL
14680      BNECHISEL
14690      BNECHISEL
14700      BNECHISEL
14710      BNECHISEL
14720      BNECHISEL
14730      BNECHISEL
14740      BNECHISEL
14750      BNECHISEL
14760      BNECHISEL
14770      BNECHISEL
14780      BNECHISEL
14790      BNECHISEL
14800      BNECHISEL
14810      BNECHISEL
14820      BNECHISEL
14830      BNECHISEL
14840      BNECHISEL
14850      BNECHISEL
14860      BNECHISEL
14870      BNECHISEL
14880      BNECHISEL
14890      BNECHISEL
14900      BNECHISEL
14910      BNECHISEL
14920      BNECHISEL
14930      BNECHISEL
14940      BNECHISEL
14950      BNECHISEL
14960      BNECHISEL
14970      BNECHISEL
14980      BNECHISEL
14990      BNECHISEL
15000      BNECHISEL

```



52,238,258,268,1,1,281,1,298,
2,76,162,2296
2298 0A7495,261,2,246,1,76,
52,238,258,261,52,268,1,76,
294,2,2026
2299 0A7498,244,184,184,74,
168,172,168,268,12,26,171,58
8,172,168,182,2456
2300 0A7499,26,171,52,226,26,
2,248,291,76,15,17,17,17,23,
89,82,1973
2301 0A750A,81,85,31,65,78,8
9,52,75,67,89,31,16,17,17,31,
896
2302 0A7502,76,77,52,76,75,7
6,67,31,74,13,15,52,67,78,68,
168
2303 0A7502,76,76,52,76,75,7
6,67,31,74,8,8,8,176,4,198,7
31
2304 0A7502,222,168,8,152,22,
1,168,48,123,252,168,8,127,2
2,248,8,2232
2305 0A7505,251,258,2,238,2
52,76,221,172,162,251,141,78
1,194,162,221,3044
2306 0A750A,171,184,84,32,3
42,171,32,88,229,168,1,168,
76,32,30,2281
2307 0A750A,162,8,168,48,52
2,252,197,52,192,194,168,48,2
1,279,194,2979
2308 0A750A,4,52,197,194,16
8,194,194,254,52,26,176,171,
192,294,172,2229
2309 0A750A,194,22,258,268,
32,162,194,171,172,194,32,15
9,194,172,192,2694
2310 0A750A,22,159,194,248,
194,168,6,22,26,171,76,168,5,
227,172,1999
2311 0A750A,161,194,194,168
2,29,227,162,194,161,192,194
174,194,194,172,2948
2312 0A750A,194,22,252,268,
31,162,194,172,197,194,32,15
9,194,172,194,2688
2313 0A750A,52,197,194,168,
171,168,182,15,26,171,32,228
222,248,261,2232
2314 0A750A,168,32,32,216,22
5,52,216,222,167,56,32,216,2
2,16,72,2161
2315 0A750A,74,74,74,261,16,
144,2,162,4,162,48,32,216,22
5,168,2239
2316 0A750A,15,261,16,244,5,
168,6,195,48,31,216,222,76,8,
8,1279
2317 0A750A,48,2,111,5,29,17,
17,82,84,82,82,84,52,82,88,7

75
2318 0A750A,82,68,82,82,21,5
2,24,15,29,17,17,68,78,68,32,
168
2319 0A750A,68,68,82,67,82,8
2,32,32,32,32,34,15,17,17,32,
73
2320 0A750A,77,84,42,76,12,6
6,67,84,67,82,32,32,32,34,
1973
2321 0A750A,54,19,16,26,34,1
5,17,17,17,75,78,82,67,82,84,
738
2322 0A750A,52,76,67,78,68,3
2,31,52,34,52,34,15,17,17,32,
168
2323 0A750A,77,84,42,76,12,6
6,67,84,67,82,32,32,32,34,
1973
2324 0A750A,54,19,16,26,34,1
5,17,17,17,75,78,82,67,82,84,
738
2325 0A750A,52,76,67,78,68,3
2,31,52,34,178,168,48,152,22,
227,1,168
2326 0A750A,123,251,167,8,222
2,294,168,1,152,252,168,8,177
2,21,248,2526
2327 0A750A,261,15,248,64,14
1,244,192,268,177,21,148,24
3,192,32,174,2541
2328 0A750A,172,247,152,168,
8,142,251,168,248,251,168,2,
1,252,251,2444
2329 0A750A,182,8,121,252,3
4,152,251,258,2,258,254,76,7
1,197,24,2229
2330 0A750A,252,182,2,121,4
3,162,254,168,8,152,46,168,8
2,268,142,1928
2331 0A750A,252,182,252,76,
228,252,252,2,252,22,76,71,
192,168,8,2621
2332 0A750A,1,247,197,192,246
1,192,261,26,244,15,41,15,24,
192,8,10,1879
2333 0A750A,36,18,141,242,17
5,76,216,192,41,12,10,18,16,
16,146,1227
2334 0A750A,192,172,248,192,
261,26,144,11,41,15,14,162,
9,142,244,2644
2335 0A750A,76,252,192,48,1
3,142,244,192,172,244,192,15
242,192,141,2229
2336 0A750A,172,76,76,14,82,
76,116,82,252,82,76,14,252,
82,52,1844
2337 0A750A,222,172,149,187
2,162,1,261,84,244,5,162,8,16
8,8,168,2194

2338 0A7A,52,166,222,168,17
8,168,192,32,58,171,562,15,1
62,157,197,2084
2339 0A7B31,242,194,192,8,28
8,2,172,149,197,261,84,268,2
42,171,149,2456
2340 0A7B77,261,68,268,66,1
68,64,16,34,2,167,48,14,21
2,167,1686
2341 0A7B88,142,32,2,168,8,1
82,8,2,152,252,2,268,264,152,
187,204
2342 0A7B88,244,168,44,151,
25,2,168,84,152,24,2,171,148
1,187,204,1999
2343 0A7B81,208,32,168,44,12
5,22,2,168,67,121,26,2,258,2
64,244,1723
2344 0A7B88,268,268,268,76,
158,194,172,152,197,268,14,1
64,8,188,8,2033
2345 0A7B82,152,26,2,268,264,
152,197,268,194,152,162,19,1
64,2,12,1521
2346 0A7B89,222,168,168,152
178,76,168,16,161,148,197,1
68,191,148,8,2621
2347 0A7B82,31,252,192,168,8,
142,8,168,48,52,211,252,176,
41,76,2626
2348 0A7B89,82,141,148,192,
148,141,148,8,32,252,192,1
68,8,152,2621
2349 0A7B81,168,48,152,252,
176,176,197,192,158,187,168,
251,32,216,229,2621
2350 0A7B76,8,32,182,258,28
8,1,76,32,211,252,168,12,16
8,187,32,2168
2351 0A7B88,171,167,182,168,
152,31,26,171,32,127,176,76,
52,228,222,2117
2352 0A7B84,252,76,142,154,
197,212,168,44,52,216,258,28
2,224,8,268,268
2353 0A7B88,174,154,182,252
1,168,157,31,216,252,262,228,
8,268,248,168,2679
2354 0A7B88,162,8,127,1,1,2,2
2,224,2,268,248,168,8,148,1
51,197,1926
2355 0A7B8A,164,52,216,222,
149,157,32,34,252,32,228,22
2,248,221,172,2023
2356 0A7B81,197,141,152,197
1,149,6,52,216,152,149,157,3
2,216,222,171,2588
2357 0A7B82,197,261,15,244,
42,261,26,268,15,192,8,242,2
67,154,168,2134

Run-stop will place an end of file marker (preprepared flag) at the position above the arrow; this point cannot be passed until it is over-written with a space or character.

Shifted run-stop will exit to the menu. Pressing any valid key will convert the original character above the arrow and then stop on to the next.

Remember in this section an up arrow (↑) denotes a return and can affect both the printer and viewfile displays.

File Length

Displays the current file's start/end address in decimal/hex and the number of spare bytes available should you wish to type in more details.

Load File

Fairly obvious this one, but it should be noted that only one error message is displayed for both disk and tape regardless of the actual error. It is "ERR04: LOAD/SAVE" and could be any fault from "file not found" on disk to the tape stop key being pressed on tape players.

The filename must be known, although the file symbol will be placed in front automatically as it is in Telcom 64.

Save File

This will ensure the current file up to the checkered end of file marker. Any alterations made in edit mode (option two) will be saved and the length of file can be made longer or shorter by moving the end of file marker. This allows one log file to be kept into shorter ones with only the required information left in them.

How to Basic

If a hex file has been downloaded (figure 2) it should be edited so that the first digit is at current position zero. An end of file marker should be placed at the end of the box.

By selecting option six, the hex will be converted to Ascii and placed at 2048 onwards to build up a Basic program in memory. On


```

3,204,1880
3300 B070045,109,44,121,250,
149,4,121,252,149,42,149,4,1
49,250,149,252
3310 B070079,115,251,145,251,
34,145,44,121,250,145,252,49
4,4,121,252,251
3320 B070145,252,251,7,208,2
27,145,251,252,175,208,220,9
4,201,201,52,2020
3330 B070175,175,252,201,49,
4,142,4,149,7,74,127,201,149
5,74,175,2020
3340 B070204,74,118,204,149,
4,142,4,149,7,74,127,201,149
4,142,149
3350 B07044,149,4,74,127,201,
149,4,142,4,149,4,74,127,201
52,120
3360 B07048,252,149,4,52,18
9,252,52,175,252,149,4,121,1
16,149,49,2040
3370 B07052,147,148,4,175,1
16,248,22,141,176,201,142,4,
52,201,252,2020
3380 B07073,194,201,52,210,
252,252,194,208,252,252,147,
74,194,201,142,176
3390 B07084,52,201,252,145,52
52,252,194,252,204,208,149,4
52,192,204
3400 B070258,74,74,51,74,75,
201,52,149,229,74,52,171,174,
74,52,1775
3410 B07048,229,145,1,145,12
4,12,12,179,201,52,47,202,12,
14,142,1420
3420 B07052,252,252,246,252,
201,121,258,9,52,50,175,52,1
5,175,74,204
3430 B070224,201,201,157,204
4,21,124,201,52,50,175,74,2
24,201,201,2274
3440 B07034,204,9,52,50,176
4,149,175,74,224,201,201,15
8,204,7,1880
3450 B07052,74,252,12,14,172
14,254,201,252,12,258,9,52
52,142,149
3460 B07053,10,192,74,224,20
1,201,158,204,9,52,111,202,5
2,10,192,1871
3470 B07056,224,201,201,134,
204,9,52,134,202,12,14,172,7
4,224,201,2444
3480 B07057,149,204,5,74,74
172,74,224,201,12,148,229,74
52,120,2017
3490 B07074,74,52,149,201,1
49,4,149,1,145,252,258,252,2
56,2,258,2210

```

```

3500 B070254,145,252,145,121
,177,145,254,141,158,197,52,
174,174,74,52,2440
3510 B07075,192,52,58,229,50
4,159,74,52,80,250,74,52,48,
229,142,1890
3520 B07084,149,1,127,225,21
7,149,74,127,221,5,202,208,2
42,149,24,2589
3530 B07091,20,4,147,1,145,
26,258,147,4,149,12,122,254,
47,204,1750
3540 B07097,20,258,244,52,2
25,201,147,47,121,252,147,27
4,215,251,175,2440
3550 B07098,46,248,14,12,140,
201,52,240,201,52,208,240,74,
142,4,1880
3560 B07099,21,258,157,244,
5,212,254,7,208,245,140,204,
147,46,12,1237
3570 B07130,171,145,175,149,
194,52,54,174,52,258,258,244
,251,74,146,2440
3580 B07148,149,1,121,42,119,
121,121,21,9,149,258,176,123,
42,5,252,2589
3590 B070258,175,121,121,7,2
60,121,10,208,221,74,145,252
,141,27,204,2440
3600 B07052,252,141,24,254,
175,27,244,24,145,24,176,175
74,254,147,2020
3610 B07084,141,24,244,24,252
,48,141,24,254,12,152,149,58
9,52,12,1741
3620 B070315,252,52,214,252,
52,214,252,52,214,252,12,214
,252,94,12,258
3630 B07072,252,147,201,145,
177,52,34,171,52,258,258,240
,201,201,44,2440
3640 B070248,4,201,48,208,24
1,141,149,147,52,48,229,52,1
2,149,149,2477
3650 B07047,549,194,12,58,17
1,52,258,258,244,201,201,49,
204,5,24,21
3660 B07049,200,201,50,258,
2,74,197,202,201,51,258,251,
145,175,201,2292
3670 B07056,145,251,24,195,2
9,141,25,204,145,252,195,4,5
41,24,204,1718
3680 B07094,142,4,145,4,137,
4,48,125,254,254,74,147,48,1
52,254,2020
3690 B07094,4,121,252,145,8,
177,252,149,4,252,252,208,2
,258,254,2711
3700 B07098,156,155,145,252,

```

```

141,17,204,145,254,141,58,29
4,56,175,12,2220
3710 B070258,252,20,140,179,2
94,173,14,204,233,4,143,16,2
84,74,240,2024
3720 B07057,177,251,175,149,2
44,194,155,224,5,201,192,40,
204,242,74,2089
3730 B070148,144,148,204,12,
34,174,12,251,202,149,156,14
8,204,12,34,2044
3740 B07074,174,17,254,175,
14,254,54,252,48,52,245,149,
147,52,12,1899
3750 B070210,252,52,240,252,
21,214,252,52,214,252,149,14
8,240,252,12,2717
3760 B07030,171,145,1,54,217
,57,254,174,149,159,237,142,7
4,142,27,1887
3770 B07034,141,20,204,52,2
82,149,149,52,21,214,252,21,
214,252,12,2254
3780 B070210,252,52,210,252,
52,210,252,12,252,252,240,14
2,241,121,204,2717
3790 B0701,74,201,152,204,24
,145,152,241,47,248,4,145,27
1,201,2,2412
3800 B070245,11,145,201,258,
2,195,252,195,251,52,171,252
,74,258,252,2487
3810 B07030,29,204,58,145,2
52,201,129,208,4,145,251,201
,5,240,25,2250
3820 B07045,252,252,50,254,
258,7,145,24,252,17,204,258
,4,245,251,2427
3830 B07038,5,254,252,12,17
1,247,74,254,201,201,148,258
,52,175,2,2444
3840 B07034,208,5,175,28,28
4,208,49,145,252,24,195,17,1
41,12,204,2044
3850 B07045,252,252,9,54,1,1
4,204,144,179,121,254,149,5,
122,252,148,2214
3860 B07057,177,252,145,1,145
,152,145,251,258,5,194,254,1
76,251,145,2440
3870 B070254,254,24,254,254,
254,145,251,258,75,254,254,2
24,147,12,194,2779
3880 B07058,145,251,52,171,2
64,52,145,241,145,4,122,179,
74,252,252,2294
3890 B070201,155,254,52,145,
252,141,24,254,145,252,141,2
1,254,147,75,2242
3900 B07048,204,12,58,171,5
49,122,144,34,254,12,121,202

```

```

,74,258,252,2372
3910 B070204,127,204,109,173
,34,258,149,79,145,251,123,8
7,145,252,155,2587
3920 B07048,175,20,254,123,8
9,175,20,254,123,74,54,145,8
8,123,70,1548
3930 B07040,5,174,12,74,92,

```

At first the edit functions (F2) view of a file may seem strange. Different boards send out an assortment of codes to make up a screen.

Figure 3
 Notes: * = return
 ---TI-COM---WTCOM 148
 50---TI-COM---
 ---TI-COM---TILCOM 84*
Figure 3a
 ---TI-COM---WTCOM 148
 50*
 ---TI-COM---TILCOM 84*

Note
 The screen layouts shown in Figure 3 and 3a will both give the same result on screen. However, when viewed, the difference is quite considerable. In Figure 3 there are TI squares each side of "WTCOM 148" then another 15 is "TILCOM 84" giving a total of 40 bytes. In Fig 3a the file only has 15 bytes as a return has been placed after the "TC" which causes the cursor to return to the start of the next line.
 As already mentioned this will differ between different boards and viewing the file is the only way to know which method is being used.

Figure 3a
Type Ctrl Z to Abort Download ---TI-COM---
 data---TI*
 * current position (4) current position 29
 Press F1 at current pos 0 and F2 at current pos 29 the file will now look like:
 ---TI-COM---data---TI*
 *

Note
 By placing an end of file marker (Z) at the point shown and then saving the file our new file will only contain "scopy".
 This procedure can be carried out as many times as required.

Joe Bradley drives

into the Plus/4's

memory and looks at

machine code

programs.

MANY NEWCOMERS TO computing may have bought a Plus/4 and have been disappointed that very few articles have been written for this machine. This article is for those new enthusiasts who wish to look into the machine and start writing machine code routines.

There is some difficulty in obtaining a full memory map for the Plus/4 but this article is intended to help in making a start.

First let us have a look at the different sections of the memory - owners of a Plus/4 are fortunate here because the computer contains an inbuilt monitor which will help.

If you wish to write machine code routines you must become familiar with the way the monitor works and the different commands that are available.

The computer contains two different types of memory location, those that you can change, called Random Access Memory or RAM, and memory locations that are Read Only Memory or ROMs, these are used by the operating system and cannot be altered.

Switch on your Plus/4 and type:

MONITOR (RETURN)

the computer will respond with

MONITOR

PC SR AC OR RR SP
; RRR RR RR RR RR RR

or something similar.

The abbreviations are:

PC Program Counter
SR Status Register

AC Accumulator
the work horse of machine code real routines
RR R Register
RR Y Register
SP Stack Pointer

This is the current address being processed by the computer. This contains six flags which give information about the current state of the processor.

This gives the next free location on the stack which is a temporary storage area used during processing.

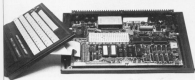
Now type:

ADDRESS 0000 (RETURN)

If a question mark is printed you have made an error in entry, probably you have typed letter Q instead of a 0 (zero numbers).

If the entry was correct you will see displayed eight rows of numbers these are the numbers stored in the memory locations 0000 to 000F. Note addresses are usually given in Hexadecimal code which counts in units, tens, 20s and 40s. Thus 000F is 15 in 16 and another is 15 (in 63 in normal decimal numbers).

Each location can hold a number from zero to 255 this is called a byte and is made up of eight binary 'bits' which can be



REMEMBER *the* PLUS/4

either zero or one. More details of this may be read in any book of machine code.

The block on the screen right in reverse print is the ASCII dump of the code if it is possible to be printed. Where a character cannot be printed it is displayed in a full stop (.)

The highest address that can be read by the processor is \$FFFF. The dollar sign shows that this is a hexadecimal number and is equal to $(15 \times 4096) + (15 \times 256) + (15 \times 16) + 15$, its decimal this works out as 65,535 which is almost 64,000 and is the reason most early computers were limited to 64K (approximately).

All the locations from \$000 to \$FFFF which are on the screen are RAM locations that you can change but only with care! The first 4K (actually 4 x 1024) bytes of memory are used by the operating system and problems can arise if you can change the value, let's examine this in more detail.

Locations \$000 and \$00C (\$3 and 44 decimal) are pointers to the start of Basic. To find these values, read down the left hand side numbers to \$000 and then move right counting 3, 4, 5, 6, 7, C. The \$09 and \$0C locations should contain the values of \$000. If you have not written any machine code then this is your

first surprise in that addresses are always stored in the computer in what appears to be the wrong order, what is called the *little (low) byte first* and then the *big (high) byte*. The pointers tell us that the current start to Basic program is \$000.

Now let us try to alter these values. Move the cursor over the 01 at location \$08 and change it to 04. Press [RETL09]. The value will now be changed and when we return to Basic the computer will take \$008 to be the start of Basic.

Return to Basic by typing a letter B and then [RETL09].

The computer prints READY and everything looks OK. However, enter a Basic line, say:

1 PRINT (RETURN)

Your screen will go haywire and nothing you do with the keyboard will bring back control. We say the computer has hung. Now we see another advantage of the Plus/4 over many other computers, it has a RESET key. This key is at the wide end to the CPU/CRT switch. Press this small button and the computer will be reset without switching off. Later when you are developing machine code routines your computer may sometimes hang but you will be able to bring your programs down if you hold down the RESET/STOP key while you press the RESET button. (A very valuable feature which you will learn to treasure).

Your computer is now reset and you now know that some parts of RAM cannot be altered without thought! As you develop your machine code expertise you will need to know which locations you can use without the consequences. The abbreviated table of memory locations from \$000 to \$FFF shows some that I have found useful, a safe rule is to restore the original value after use.

Consecutive locations in the region \$0000 to \$00FF are particularly valuable - this area is called Zero Page because the H byte of these addresses is zero. Consecutive bytes in zero page are often used as pointers in machine code routines e.g. LDA (LDR) tells the computer to look at \$00 for the H byte of an address, look at \$0C for the H byte, add the value of the Y register to the address obtained and then load the accumulator with what is found at the calculated address.

We have seen that Basic normally starts at \$0001 but if you type

GRAPHIC1 (RETURN)

The screen will show a hatched pattern because you will have moved into the high resolution mode. This mode takes an extra 1K of RAM memory and the computer gets this by moving the start of Basic up to \$0001. Even though you may not be able to see on your screen

System Memory Locations

ADDRESS	HEX	DECIMAL	SYSTEM USE	AVAILABLE FOR USE
\$0000-\$000F	0-15	0-15	Input output chip	NO, labels alone
\$0002-\$0006	2-7	2-7	Temp. used in search and remainder routines.	YES
\$0008	8	8	Flag word in quote scan	YES
\$0010-\$0015	16-21	16-21	Temp storage evaluation	YES
\$0018-\$002C	24-44	24-44	Start of Basic	YES - last reset
\$001A-\$0020	26-32	26-32	Start of Basic Variables	YES - last reset
\$001F-\$0028	31-40	31-40	Start of Basic Arrays	YES
\$001A-\$0020	26-32	26-32	End of Basic Arrays	YES
\$0030-\$0034	48-52	48-52	Bottom of strings	YES
\$0035-\$0036	53-54	53-54	String pointers	YES
\$0037-\$0038	55-56	55-56	Top of available memory	Only move down to protect memory
\$0039-\$003A	57-58	57-58	Current line number	YES
\$003B-\$003C	59-60	59-60	Pointers used in get character routine	YES
\$003D-\$0042	61-66	61-66	Pointers in ROM routines	YES
\$0043	67	67	Graphic mode	NO
\$0044	68	68	Colour Selected	
\$0045	69	69	Mathematical I	
\$0046	70	70	Foreground colour	
\$0047	71	71	No. of columns - screen	
\$0048	72	72	No. of rows	
\$0049	73	73	File length	
\$004C	76	76	Logical file number	
\$004D	77	77	Secondary Address	
\$004E	78	78	Device number	
\$004F-\$0050	79-80	79-80	Pointer to file name	
\$0051-\$0052	81-82	81-82	Pointer current screen line	
\$0053	83	83	Cursor column	
\$0054-\$005F	84-95	84-95	Area used to store more after number conversion	
\$0060-\$006F	96-103	96-103	STACK	
\$0070-\$007F	112-119	112-119	Custom tape buffer	Very useful to store short machine code.
\$0080-\$008F	128-135	128-135	Logical file numbers	
\$0090-\$009F	144-151	144-151	Primary device numbers	
\$00A0-\$00A9	160-169	160-169	Secondary addresses	
\$00B0-\$00B9	176-185	176-185	IO Keyboard buffer	
\$00C0-\$00C9	192-201	192-201	Monitor control for RAM/ROM	
\$00D0	208	208	Start of Basic Text	
\$00E0	224	224	Start of Basic Text when \$0025 is being used.	

which key you are pressing, carefully type:

GRAPHICS[RETURN]

and you will return to the normal screen.

So we see that the operating system takes values \$1000 (approx 4K) at the bottom of RAM for normal Basic, or \$4000 (approx 16K) for high resolution graphics.

What about the top of RAM? Well, apart from a small area from \$7000 to \$71FF which is again used by the system, the rest of RAM is available for Basic programs. The amount of memory from \$7000 to \$7FFF is \$8017 and this is the number that appears on the screen at line power-up.

However, to work all the Basic system of the computer, it needs another 12K of memory - the ROM. Where does this go - we already know that the computer can only read 64K of memory and it looks as though this is all taken by RAM. What happens is that the ROM for the operating system has addresses from \$0000 to \$FFF. So there are two different types of memory that have the same address one byte in RAM and another byte in ROM. When the computer is working it needs a switch between RAM and ROM so that the correct byte is read.

Different areas of memory are called memory banks and we need banking routines to switch different banks in or out.

The Plus/4 memory map may be illustrated by Diagram 1.

Diagram 1		
Memory Area	RAM	ROM
\$7000 - \$71FF	Banking Routines	Operating Routines
\$0000 - \$7FFF	Available for Basic Operating Routines	
\$4000 - \$7FFF	BASIC AREA	
\$1000 - \$3FFF	Either BASIC or HIGH RES. GRAPHICS	
\$0000 - \$0FFF	RAM used by system	

When first switched on the computer has access to RAM memory bank \$0000 to \$7FFF and ROM \$0000 to \$FFF. However the P016 and P00E commands will always access RAM. A machine code program could be P00ED to the top of RAM (below \$7000)

but a system call to this part of memory would result in a ROM routine being entered at an unusual point and execution continued from this point. (Execution usually means data at the computer in this case - it will probably hang!)

The in-built monitor ROMMON allows you to display memory locations or disassemble either RAM or ROM. Let us try some examples. Type and enter the Basic program:

```
20 SA=992
20 PC000=000
30 ROMADR=P00E($A41),1
40 NEXT
50 DATA 128,128,128,127,200,12,
202,208,209,96
```

This is the type of program that appears in this and other computer magazines. The numbers in the data statement are P00ED into memory one by one starting at location 992 if the program is ROM.

The simple machine code program following will be entered at \$0000 which is in the Tape Buffer area and thus safe from corruption.

```
0000A21A 128 P016
00020A 128
000300C0C $1A $00C0C
00041A 016
000700F9 016 $00E
000900 016
```

Let us look at this in detail. The first command loader sets the X register to \$1A which is the same as 26, the number of letters in the alphabet. The command 128A transfers the

six 127 because the initial position corresponds to X=0. The 128A instruction is like a P00E, so a letter Z will appear on the screen. The next line decrements X by one which now becomes 25. This is not zero so the B016 (Branch Not Equal) sends the processor back to the 128A. This will result in a Y being printed and so on until X is zero when the program will go to the R15 (Return from Subroutine) and return to Basic, so all the letters of the alphabet will be printed in reverse order. \$15 992 (P016,128) will run the routine.

OK, so far so good. Now let's try the Monitor. Type MONITOR and [RETURN]. Actually M and (shaded) C mean abbreviation that could be used.

Then type:

```
D 0000 $000
```

when the above machine code should be displayed.

Let us try to move this to an address where we have both ROM and RAM - above \$4000.

Type 5 and [RETURN] to return to Basic. List the program already entered and change line 10 to read 5A=128M. It is a good idea to save this program now. The number 128M is the decimal equivalent of \$0000, now ROM the program. The machine code will now be in RAM from \$0000 and it might seem that \$15 128M would run the program. Well try it! What happens is that the \$15 call goes to the memory bank that is switched in, which is ROM above \$0000 and happens to have a routine to give a warm start. If you saved your program then reload or type it out again. ROM to make sure the machine code is in RAM.

To check, go back into the monitor by typing M (Shaded) and [RETURN], then D 0000 (P016,128).

What you have seen is ROM and not our little program.

However the Monitor is controlled by location \$0016.

Type M \$710 B017 to display memory locations, and type over the first 80 with 80 then [RETURN]. The monitor will now display RAM above \$0000. Type D 0000 again and there should be our little machine code program. You should note that although the monitor is displaying RAM, if you used the C command, i.e. C 0000,

then ROM would again be entered. (Don't do it!)

Now can a machine code above \$0000 be executed? The answer lies in two memory locations.

```
MPH = ROM select
MFM = RAM select
```

Any write (i.e. P00E) to MPH will select ROM and any write to MFM will select RAM. However care is needed. If you do P00E to MPH the machine will hang! This is because at the first interrupt the interrupt vector will point to RAM and run the usual interrupt routines in ROM. Thus the interrupt must be disabled before a switch to RAM and then cleared after the call to RAM is finished.

Type the following extra lines:

```
60 SA=992
70 PC000=011
80 ROMADR=P00E($A41),1
90 NEXT
100 DATA 128,141,141,200,12,
0,128,141,141,201,88,96
```

Now, if you RUN the program in addition to entering the old routine at \$0000 in RAM you will also have entered the following routine at \$0016:

```
000070 001
000103B0F $1A MPH
000120 0000 $B $0000
000103B0F $1A MPH
000150 C11
000160 R15
```

You could check, by going into monitor and disassembling from \$0000 and also \$0016. The command \$15 992 (P016,128) will run routine.

Let's examine this in detail. When you enter the command \$15 992 the program will go to \$0016 and set the interrupt (i.e. the interrupt no longer occurs). The next op-code at \$0017 looks as though we are trying to put the value of A into the location MPH but actually this acts as a switch which changes from ROM to RAM, the next op-code makes the processor jump to the sub routine in RAM at \$0000 and executes it - printing out the alphabet as before. The R15 at the end of the \$0000 sub-routine returns the processor to \$0007 where the 128M B016 switches back to ROM. The interrupt is returned to normal and the final R15 returns us to Basic.

GRAPHICALLY SPEAKING

Stuart Cooke takes a close

look at Vidcom 64, a new

low priced art package.

NO MATTER WHAT SORT OF PROGRAM you are writing for your C64, presentation is important. If you are writing a business package then the screen should be made to look as interesting as possible. On the other hand if you are writing a game you will need to provide an interesting backdrop for your game. There's no point in writing the best ever platform game if it isn't pleasing to the eye.

Designing screens on the C64 is not that easy since there are no built-in graphical commands available. Therefore, many programmers will use a graphics package that will help them draw 'pictures' in as short a time as possible and with ease. Numerous packages and peripherals are available. For example you could use a light pen or a touch tablet or even your joystick.

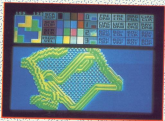
Vidcom 64 is another package to add to that already overflowing number of programs available. It does however have one feature that will make it stand out above the others, its price is only £4.95.

Vidcom's 64 will work on either the C64's multi-colour or standard bit map screens. Standard bit map mode allows you to use two different colours in any character square on the screen while multi-colour mode allows you to have four colours in any square with a loss in horizontal resolution. I.e. the coloured 'dots' are twice as big.

The best controller to use with Vidcom 64 is a trackball, however for those who can't afford one of these fairly expensive devices, a normal joystick will work just as well.

Layout

On entering the program the screen is split into two halves. The upper half is the top half of the screen on which you will draw, the other half is the control





panel which displays most of the functions available. I say most as there are actually three menus which appear at this position, each one being selected by the function keys. Allowing your controller will move a small arrow around the screen allowing you to choose which command you want to use. Each press of the first button is acknowledged by a tone and the command that you have selected starts to flash.

Entering the drawing area is simple, you just have to press the control key. As I have previously said you can only see half of the drawing screen at once. However, if you move your pointer down the screen the command menu will flip up to the top of the screen allowing you to alter the contents of the bottom half of the screen. Sometimes this is quite difficult to use as you can't see what is on the other half of the screen to which you are writing on. Thankfully the author of the program has included a way of turning off the command menu allowing access to the full screen.

The three command menus available are the drawing menu which allows you to select all of the drawing commands, the definition menu which allows you to manipulate sprites, characters, and patterns and the L/O menu which allows you to save all of your work to disk or tape.

Commands

All of the expected drawing commands are available, the have the freehand draw, hand draw, line, circle etc. There are even some extra ones such as the ability to duplicate areas of the screen, rotate the contents of a box on the screen and reverse the contents of a box.

A few fill commands are also available from the drawing menu. Velocim 64 has two graphic screens available for use. The merge fill routines available in this menu allow you to merge areas of one of the graphics screens with the other.

A 'normal' fill routine is available from the definition.

The PAT FILL command allows you to fill areas of the screen with a pattern of your choice. This can be one of the patterns included in the program which consist of everything from a solid to a brick wall, or you can use the grab definition command which allows you to generate a new pattern by grabbing an area of the screen.

As you are no doubt aware, the screens in most games programs are not saved as a picture. Rather, the screens are built up from a number of re-defined characters. The reason for this is that using characters will take up a lot less memory and you can use characters

from one screen on another, saving even more space. With Velocim 64 it is possible to grab character definitions from the screen. This means that you can use this package to design your screen and then save it as a number of characters. You can use these characters to design your other screens.

Not only can you grab characters but it is also possible to grab sprite definitions. Now it is a simple matter to change an area of the screen into sprites for use in another program.

Input/Output

The L/O menu offers a wide range of functions. It is possible to LOAD, SAVE, VERIFY and REPLACE data. Data can be the actual graphics screen, or the patterns, character or sprite definitions. It is also possible to select whether you are using cassette or disk from this menu.

Gripes

Obviously no program is perfect and I did think of a few improvements that could have been made to this package to really make it stand out from the rest.

There is no function that allows you to get a picture dump of the screen that you are designing. A screen dump is often quite useful for reference without having to load the screen back into the computer. There are many similar packages that do offer this facility.

I previously mentioned that the program makes a beeping noise whenever you select a command from one of the menus. However some of the drawing commands need more than one press of the first button to operate. No indication is given when the program has acknowledged the first press. From experience this quite often means that you end up with circles and boxes that you can't use. A simple beep after each press of the first button would have made things a lot clearer.

For 14.95 it is very difficult to fault Velocim 64. It is an extremely easy to use and powerful program offering many facilities that are only available on more expensive programs.

If you don't own a graphics program then I would suggest that you go out and buy this. Even if you can't draw its great fun just messing around.

If you already own a graphics package then this is still worth looking at as it has some very powerful features.

Touchline

Name: Velocim 64
Supplier: Activision
Address: 23, Pinedale St, Harpenden Herts AL5 2PN
Price: 14.95

THE MONSTER RETURNS

[illegible]

[illegible][illegible][illegible][illegible]

BOUNCES

Amiga 1.04 (1991)

10 1 6 5 4 8



THERE HAS BEEN A LARGE number of "fighting" programs appearing over recent months, we've hadkung-fu, wrestling and shooting. Now Beyond has brought out a game that combines the elements of a fighting game with those that expect you to get a ball in the goal, as in football.

You find your player in an arena, you at one side and your opponent at the other. Both players are tied to the wall behind them by a length of elastic, and are armed with some sort of scoop with which you can try to catch the ball that is flying around the arena.

Points are awarded for getting the ball in the net, which is situated at the top of the screen or for knocking your opponent over.

What makes this game so much fun is watching your players floundering on the floor being dragged along by the elastic and seeing them being knocked senseless by the walls at either end.

There really isn't a lot more that can be said about this game apart from the fact that the idea is extremely original and that Beyond has created a game that is great fun to play.

J.C.

TWO JINJA

Phil 17.95 joystick Required

7 1 6 8 4 7



ONE OF THE bloodiest battles in the Second World War was the attempt by the US Marines to regain the island of Iwo Jima from the Japanese. Before we get to the title is the Phil Wargames Series, you too can see if you have a fair for tactics.

You have between 12 and 16 turns to eliminate all the Japanese forces from the island depending on which of the five skill levels you select. Tactics must be selected carefully as both all out attack and solid defence are likely to meet with failure. While you receive no defence merits (weather permitting), the

Japanese don't and so tend to launch suicide attacks when a unit is close to being wiped out.

The game is controlled entirely via a joystick using it to select from your various units and determine their options from ordered menus. These options include looking at one of the six beaches, move, attack or pass. Each unit has aggression and defensive factors, a movement allowance and a combat range.

Two Jinja is a very user friendly wargame and I found it more challenging to play than its sister game Fallahajj SE.

G.B.H.

KONAMI'S PING PONG

Imagine 19.95

8 1 4 8 6 7



IN THE IBM AND ORIENT past the only video game available was a crude table tennis simulation called Ping Pong. Do not confuse it with this little gem from Imagine.

The opening screen is typical Japanese screens where a ping-pong ball bounces out Konami's name and then bounces down to the head of an unimpressive table. After selecting the level of play the next full graphic display shows the view of the table from the human player's end with an audience just off to the wings. At each end of the table is a streamlined hand grasping a bat.

During play the bats follow the path of the ball automatically and the only control that you need worry about is the kind of stroke to play.

Occasionally a stroke will result in a "Buster", a weak shot which suggests a superior smash at a table. In the lower levels this is no problem but as you progress through the levels, the game gets faster and faster and when there is no reflex action just like the real thing.

The only quibble about this excellent conversion is the lacking appeal of the game.

102

SKYHAWK

Bug Byte CDSR C-16



DURING THE CURRENT burst of interest in the C-16/Play-It market a lot of C-16 games are gaining a new lease of life. Skyhawk resembles Virgin Games' Falcon Patrol which has long been a favourite of mine.

As the pilot of an advanced NGC fighter it is your job to see off the enemy bombers which are devastating the countryside. Fuel and air-to-air missiles (AABs) must be monitored carefully during each battle and should the need for re-equipping your fighter arise you will have to be fast and careful and keep an eye on the radar

for approaching bombers.

At first each wave consists of two planes but this gradually increases to four as the game progresses.

Unlike Falcon Patrol, the jet cannot descend into the 3D scrolling landscape. This makes retreating less hazardous, but still risky, because the plane cannot crash unless it collides with a bomber.

Skyhawk is a game of survival. The bombers keep coming until you have run out of jets or fuel after all of the retreating platforms have been destroyed. Cheap, cheerful and quite addictive.

I.D.

BANDITS AT ZERO

Mail 12/91 C-16 - jaywick



all colour as the daylight gradually fades. By now your fuel is critically low but there is help at hand as you rendezvous with the tanker once more. Tacky business, this aerial refuelling lark. Altitude and speed are critical in the umbilical cord is attached and if you take too long claws will break and the tanker will retreat left to safety. Lowly attacks are few at night but occasionally a fighter will appear. The best policy is prayer... Fuel is more important than fighting as with a little bit of Divine Intervention and a gaming of your teeth as you bite the cord, the night will pass without the loss of another shield and the sun will come up on a freshly refuelled jet eager for another day's action.

The new day brings with it a new breed of pilot, the rat gun. Borne in your face and their rockets spit out at you if they get behind your plane. More jets appear to be flying towards you at high velocity and your shield now is in danger. Night seems a long way off as you fly on towards your target across the never-ending sea.

Another nocturnal refuelling finds you closer still to your target. This is indicated by the fleet of tankships which are sending up a battery of shells. Still the enemy planes attack and the tiny black flecks of fish threaten to blot out the sun.

Before long you begin to wish that day three had not drawn and repeat that it did.

Graphically, the game appears a lot more sophisticated than it actually is and in comparison to many other C-16 games it shows what a thoughtful programmer can do within the cramped memory confines of the machine. Adding an extra problem to deal with as each screen goes by, helps to ease all the boredom of a night forward shoot-out-gp.

The only really weak point of the game is the music which is monotonous and unimaginative. It only appears at the end of each day if you don't complain too loudly.

Ammunition supplies are reminiscent of a cowboy's lagoon in the old west movie days. Bullets are unlimited.

In this way the temptation to fly with trigger finger down would have to be resisted in favour of accurate and thoughtful gunnery skills.

Don't allow those criticisms to draw you away from the fact that this is a superb game worth every penny of your hard earned cash. If things do get a little dull you can rescue your jet back and forth across the sky.

In this way the program contains more money games than flying towards you at a high rate of knots increasing your problems threefold.

I.D.

AIR ATTACKS FROM amphib carriers can best be stopped by sinking the floating airship which harbours the planes. Good logic but not as easy as it seems. First you have to battle through the waves of fighters which protect the ship. This is your mission in Bandits at Zero and it is quite a challenge.

The game starts with your solo fighter-bomber flying low over the sea searching for the enemy. First there is an encounter with a retreating plane to top up your tanks ready for the long day ahead but soon your radar display becomes live with tiny blips which denote the enemy fighters ahead. As the plane flies on to join combat the screen scrolls smoothly as a coastline

unscapes is gradually unveiled, but this is no tourist trip because soon the enemy is there in front of you.

The first day is also pickings because most of the planes are flying in the same direction as you so you can shoot slowly. Then by surprise, the planes turn fire back and the only real problem is your own skill at dodging collisions with them. A cautious eye should be kept on the radar screen for the occasional attacking plane which comes in on you from the opposite direction at high speed. If a collision occurs you lose one of your armor protective shields.

After this leisurely cruise across the rolling seascape, night gradually falls once you

TIGERS IN THE SNOW

IBM Gold 1992

6 1 8 5 7



IN DECEMBER 1944 HITLER'S army was being driven back towards the Fatherland and a major counter offensive was needed to drive the allies away from the Rhine. The conflict that ensued became known as the Battle of the Bulge and marked the beginning of the end of World War II.

A major part in the offensive was played by the Panzer Tank Divisions of the German army. These powerful tanks had earned themselves the nickname of "tigers", hence the title of this simulation.

As in the real battle the action begins on 16 December and ends on 27 December. Players can take charge of each army or the computer will play the role of the enemy.

The weather conditions of these fateful days of long ago are faithfully reproduced during the game, determining whether your airborne supplies can be flown in or not. Being Northern Europe in the dead of winter, you soon learn to take advantage of the few clear days that occur. Supply level affects the fighting strength of your units and can become a crucial factor in your success.

The situation at the beginning of the game is that the Allied forces are represented by greenheads. American troops against a very experienced and strong German army. The initial aim of the Allies is to hold back the onslaught until reinforcements arrive. For the Germans it is to break through and cross the River Meuse, taking the

major towns of Bastogne, Marche and Kochem on the way for maximum points. The eventual aim of the Allies is to breakthrough to the east.

Each day is broken down into a sequence of five moves. Firstly, the Germans alter their positions and enter into such decisions as their moves allow. Next it is the Allies turn to do the same. After this, reinforcements are placed on the map and Victory Status is assessed to see if the game can continue. Finally, you are given the option to save the game before the next day dawns.

The result of a battle is determined principally on the relative strength of the units involved and the strategy chosen for attack and defence. If the result of a battle is a dramatic win, the victorious unit may advance three squares and attack again later that day. Defeated forces may retreat if there are any combat points left. Sometimes this will result in under-supplied units and they will be wiped out.

Attack strategy can be a major offensive, a medium battle, a light skirmish or a gnat bite of a reconnaissance mission. In defence you may choose to counter-attack, try to hold your position, withdraw or try delaying tactics. Choosing the correct response to your opponent's strategy is the aim so that you will lose fewer combat points,

if any, and live to fight another day. The longer a unit survives the greater the number of combat points that are awarded at the end of each day. A unit with fewer than 25 points is in serious trouble and must rely on plenty of movement points to pull back out of harm's way while they recover. This I learned from bitter experience!

Extra combat points can be awarded if you choose to include an artillery barrage with your attack. Each day you have a number of artillery points awarded and you are given the option of using a limited amount of this power to shore up a flagging unit.

The documentation which accompanies the game is extensive but confusing. A map of the whole area is also supplied but I found it to be confusing. Locating my starting position took a while and this wasn't helped by the key being some distance from the map itself.

Eventually, I struggled through the documentation and found that the best way to learn is through experience. I fought long and hard but I regret to say that if it had been left up to me, I'd all be sleeping. German tank units fly round. This is an enjoyable simulation but the battle is popular for computer strategy games and I have seen better in my time.

R.D.

RING COMMANDER

Macintosh 1.1/7

3 4 6 6



YES, IT'S YET ANOTHER flight simulator! This one sees you in charge of a jet fighter defending your island from the enemy planes. Their target is the nuclear power plant and it is vital that you intercept their bombers before they reach their destination.

The screen display is in two halves, the top window showing the view from your cockpit while the bottom part of the screen contains your instruments. These include a radar, compass, several gauges and warning indicators and a scrolling map of part of the island. A full screen map of the entire

island can be called up at the press of a button.

Although the instruments look complicated, controlling your plane is very simple. You can attack an enemy plane with either cannons or missiles, but you have finite supplies of both. You also get the chance to refuel in mid-air providing that certain conditions are met.

Ring Commander was written back in 1984 and is beginning to look a bit dated now but if you want to try a cheap flight simulator before splashing out on a more sophisticated model, it could be just the thing you are looking for.

C.M.

OFF THE HOOK

Hearts Dreams £19.99 C&A



PRODUCING COMPILED tapes for charity organisations is becoming popular. Off The Hook is yet another and all proceeds from the tape are going to the Prince's Trust for the rehabilitation of drug addicts.

It is very difficult to comment on a tape that is being sold for charity as you obviously want the people involved to sell as many tapes as possible. Thankfully in the case of Off The Hook the 18 programs included are all of a fairly high quality.

Programs included on the tape are the second of Activision programs featuring Pitfall Harry and a great fun-to-play Harry is trying to rescue his niece - makes a change from Princess - while avoiding the many dangers on his way such as scorpions,

giant frogs and electric eels. This game kept me occupied for a long time, never does my have something to do with this!

Next on the cassette is Space Pilot 3 from Activision. This game offers nothing out of the ordinary and looks a little dated. Nevertheless, flying your space ship over a scrolling backdrop and shooting enemy ships is still great fun.

Probably the largest and most complex game on this cassette is Pytron from Beyond. When this game was launched on the Spectrum around two years ago I didn't get anywhere with it and I've not improved in those two years. You arrive in charge of the Heals II installation and must defend it against enemy invaders. This means that you

must shoot them out of the skies, go after any that get into your buildings and use to any traps that your complex needs. Pytron is a very complex game - the fact that the instructions take up more space than all of the other games on the cassette put together shows this.

Tell Guy from Time allows you to play that famous TV character. Col. Seavers the nut man. Your job is to perform all the stunts seen in as little time as possible. Nothing really exciting here but well worth loading in for a quick half hour.

Demon of Torgar from Firstland places you on the asteroid Torgar. Your aim is to find the sacred crystals that are scattered around the many levels of the asteroid. First, you're puzzled if it. Demons of Torgar is a platform game. It is however quite well produced and is good fun to play.

Steep in Space from Harman should need no introduction. But, just in case you've never heard of it, it's a little similar to a Defender game, but in this case there's no one to rescue and you must kill all of the Hostiles before they can blow up your planet. Oh by the way, you aren't given a spaceship, this time you are an intergalactic sheep.

King Solus. Back from Ocean is probably the most

disappointing game on the cassette. Mind you, when you think of the price it's not all that bad. In this game you must guide your man up the fairground track to rescue your damsel in distress. Of course there are objects for you to avoid on your way up such as roller coaster cars and springs.

Black Thunder from Quakeba finds you changing along a number of roads shooting everything in sight. An extremely fast game and great fun to play.

Death Star Interceptor from System 1 is the penultimate game on the cassette. This game finds you once again in control of a space ship trying to save the galaxy. Again nothing out of the ordinary is offered but if you are into tapping alloys then you'll probably enjoy it.

Bringing up the rear is Talkings, and that's exactly what I did in this racing game, and this one offers nothing out of the ordinary though it is great fun to play.

When you consider the amount of money that is being asked for this cassette and then work out how much is being asked for each game it's impossible to complain.

Even if there are only a couple of games on the cassette that you haven't already got then it is still worth the asking price. Go on and buy it NOW. SC.

CALEDRON II

Palace Software £7.95 Joystick required



ANYONE WHO DEFEATED the evil pumpkin in the original Caldrun now has the chance to run the tables. The Witch Queen is in charge of a wholly evil regime. The only person who can put things right again is you - a brave pumpkin warrior!

Caldrun II is a sort of platform game but instead of jumping, you bounce and this is likely to take you some time to get used to as our hero rebounds off walls like a ball bouncing on a pin table. Tearing in one of several randomly chosen locations, you must collect scattered objects before you can defeat the queen. These

include crows, swords and ax. There are various nasties trying to stop you and colliding with them depletes your energy. You can however pick up glowing spheres which both replenishes your energy and allows you to fight back by hurling these spheres at your enemies.

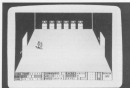
The graphics are good and amusing with some lively touches such as when a gargoyle tips you off a platform into open space. But I would have preferred it if everything scrolled instead of jumping from screen to screen - you are never quite sure what nasties lie in wait for you. G.M.





COUNTDOWN TO MELTDOWN

Matteotronic MAD Range (£2.99 joystick required)



AFTER AN EXPLOSION IN A nuclear power plant, the central core is overheating and there is a considerable danger of a nuclear blast. Your task is to dampen down the reactor rods using a team of remote-controlled android commandos.

You have eight robots to manipulate, each with a different set of skills and must guide them through the 2000 rooms on eight levels as you try to reach the core in time. There are assorted intruders that must be disposed of before they sap your strength and mobility etc. and render that particular robot useless

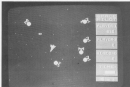
until he can be repaired by one of his companions.

Moving the robots and firing is via the joystick but there are also several keyboard commands for using the objects that you have found. The graphics are fairly simplistic - 3D views of the rooms with exits and enemies shown but you can switch between views to show the doors more clearly.

Countdown is an interesting game that will keep you quiet for ages - you can save your current position. If you enjoy large scale mapping games, this is good value for money. **C.M.**

SPACE PILOT

Amos/Clugsoft C-16 • joystick £3.95



SPACE PILOT STYLE GAMES have been with us for many years but this is the first that I have seen for the C-16.

The game is one of survival as you roam the galaxy facing the attacks of innumerable alien craft. Flying singly or in formation these demons are armed with heat-seeking missiles which will attempt to snuggle up close to your ship, vaporising you as they do so.

The background of stars swirl with a beautiful 3D effect in every conceivable direction but this is where the price of sophistication must be paid. All this activity results in a dazzling effect

on the spaceship's movement lowering the excitement which the gameplay has to offer.

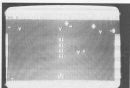
It is also noticeable that the complexity of the angled swirl is slower than the simpler vertical or horizontal swirl but I will accept this as a game 'feature'.

I do feel that the game is playable and certainly stunning. Each of the four screens of aliens are armed with missiles which behave in their own characteristic way.

To use a cliché, this is a flawed masterpiece. **R.D.**

SOLO

Bug-Byte C-16 • joystick £2.95



PROGRAMMER STEVEN Koller's name keeps cropping up on C-16 games. He specialises in unapologetic shoot-em-ups which rely purely on fast reflexes for survival.

This time he appears on the Bug-Byte label with a typical product of his endeavour which provides mayhem with a storyline.

Solo is the fighter you control in your battle against the aliens who appear in droves to annihilate you. The action is swift as they haul everything at you in their omniscient attack. Solo can move in any direction on the screen.

At the beginning² your craft sits on the bottom border but you'd be foolhardy to sit there too long. In a matter of seconds the screen explodes into frenetic activity and a second wave of attack will start before you have wiped out the first.

There are 10 waves of terror to test your staying power and I suspect that survival through all of them is impossible unless you have nerves of steel and reflexes honed to superhuman efficiency.

Lovers of unpretentious

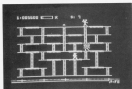
sapping games will love this. **R.D.**

➤ ACTION REPLAY



THE CHIP FACTORY

Supernint C-16 • Jovetick 16.95



CHARLIE IS A TRAVELLER accountant who sneaks into the chip manufacturing plant to try his skill at production. Bugs, sparks and the odd spinner in the works run rampant at night and it's up to you to guide the ball around the screen to produce his microchips.

This is a platform game and a conveyor belt runs along the bottom of the screen. On the conveyor are sockets and Charlie has to drop the chips from level to level so that they fall into an empty place in the belt. When all of the places on the belt are filled he must climb

to the switch at the top of the screen to wind the belt on.

The enemies also wander around the platform and contact with them can be fatal.

Completely filling a conveyor belt allows Charlie to move on to a new production line and further problems.

The graphics could be more imaginative but this does not affect the game too much. The action is certainly frantic and the planning of your next move is frequently thwarted by the numerous snakes. Simple but effective.

J.G.

HOCUS FOCUS

Quicksilver 16.95 Jovetick optional C64



EREMY IS A SDF PHOTOGRAPHIC working for the Daily Muckler and is given the assignment of taking pictures of the Potty Professor's inventions in his hidden lab. The whole cave system is radioactive and protected by genetic mutants who are intent on stealing your film and roasting - your one means of protecting yourself.

The parts of the inventions are hidden in assorted objects which must be searched before you can take a picture of them. Some objects contain ghostlike figures that again steal your film. When you have some pictures ready

to develop, you can make your way back to the start where you get the chance to place the pictures on to a large grid.

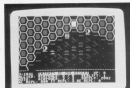
Moving Jeremy is a simple left, right and jump although there are several other functions which are icon driven. These icons examine an object, take a photo, pick up or switch on an object and fight.

Icons are supposed to make things easier to manipulate, but they just don't work here.

C.H.

KNIGHTS OF THE DESERT

US Gold/SH CTSI cassette, CTSI disk C64



DESPISE THE FACT THAT HE was on the "other side" during the Second World War, Rommel was respected and respected for being a decent fellow and a brilliant commander. It was only the fact that the British, under Montgomery, had cracked the German codes and so knew every move that he was going to make, that led to his ultimate defeat in North Africa.

The first thing to be said is that this is an incredibly complex wargame and would be best suited to someone with a fair amount of experience or a lot of patience. Players take it in turn to rein-

force, supply, move and fight as the Germans attempt to take Alexandria whilst defending their main base of El Agheila.

The game is played on a hex grid that while not exactly graphically stunning, is perfectly adequate. The instructions are long and complex (a book would have been better than the huge closely printed sheet) and the gameplay is hardly user-friendly. But, if you enjoy wargames or are interested in the historical aspect, there is an awful lot here to keep you occupied.

C.H.

[illegible]

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

YOUR COMMODORE IS ALWAYS ON the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together?

If you have written a program, be it a game or a utility, that you think you should share with other Commodore owners, then why not submit it to us for publication. Don't forget we pay very well for any articles that we publish.

All submissions should be well documented and where possible include a line by line breakdown of the program together with a list of any variables used. A copy of the program should be included on either tape or disk.

All submissions should be sent to the address below. If it is not chosen for publication then it will be returned to you.

You may not have written any software yourself, but you may have very firm opinions about the world of Commodore and all its attendant industries and products. If you do, then put your views or questions on paper and post them to us again at the address below - you might even get paid for airing your views!

All submissions should be sent to:

The Editor
Your Commodore
Angus Specialist Publications
No 1 Golden Square
London W1R 3AL

Office Use Only

Surname

Forenames

Address

Postcodes Telephone Number:

Times to contact point

Program Name:

Computer/memory slot:

Status required (disk etc):

Have you sent the program to another magazine? Yes/No

PLEASE COMPLETE IN BLOCK CAPITALS

Date received: Date ack. sent:

Date checked: Checked by:

Rating:

Comments:

Reply sent: Ack. received:

Date printed: Date returned:

Fee:

Date Fee sent:



• I N C E N T I V E •



THE GRAPHIC ADVENTURE CREATOR

The amazing, combined text and graphics adventure generator enables you to produce fast, compact adventure games with the minimum of effort and the maximum of creativity. Stunning graphics can be created quickly and easily with the powerful picture generator that has many features including - dot, circle, ellipse, line, ellipse, fill fill, shading, step by step review, easy editing and the ability to merge pictures.

The comprehensive, intelligent command interpreter can handle complex sentences and multiple input commands. There is also an extensive text compression facility that allows you to produce far more detailed and numerous location descriptions. Plus a full function editor, automatic word formatting, a logical command interpreter and an abbreviated input acceptance facility.

So unleash the power of your imagination now with The Graphic Adventure Creator for your Commodore 64 computer. Available from all leading retailers or direct from Incentive on 0734 598678.

INCENTIVE SOFTWARE LTD. 54 London Street, Reading RG1 4SQ.

Please rush me, for my Commodore 64 Computer **THE GRAPHIC ADVENTURE CREATOR**

☐ Cassette £22.95

☐ Disc £27.95



I enclose Cheque/PO for £ _____
or please debit my Credit Card

No. _____

Name _____

Address _____

George Duxal has managed to get hold of some Amiga games software. Read on to find out what's available.

IT IS A REAL SHAME THAT Commodore, in all its wisdom, has decided that the Amiga should be a business machine. When you think about it, with amazing graphics, 4096 colours, stereo sound and a 68000 processor, there cannot be a machine more suited to high-quality games.

Fortunately, many of the large American software houses agree with me, and slowly but surely some excellent products are appearing. Electronic Arts was the first company to produce games for the Amiga, not surprisingly perhaps - Commodore gave development machines to Electronic Arts more than six months before anyone else!

Most EA owners will recognise the first three games EA released - Archon, One-on-One, and Seven Cities Of Gold as they are all conversions from C64 originals. None of these products take full advantage of the Amiga's capabilities, but each has its own touches that make it just that little bit better than anything available for a normal home micro. But then the Amiga is no 'normal home micro'!

Seven Cities Of Gold was the first complete game for the Amiga, however it is also the least good. It is ridiculously similar to the 64 version, even down to the chunky horizontal scrolling which is more reminiscent of the 16:10 than a 17:1000 Amiga!

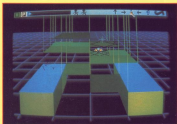
In The Basket

Things improve greatly, however, with One-on-One. Although it too is a straight conversion from a well known 64 game, the graphics are good and the sound is absolutely mindboggling! In case you don't know the game, it's a basketball simulation, except that you don't play a whole team, just one player - Dr L, or Larry Bird (both of whom I am assured are well known basketball stars). You must try and out-fox your opponent and score as many points as possible in the time allotted.

Graphically One-on-One is good, though by no means special. By using vertical sprites, the programmers have taken the map way out. Had they used the much vaunted 'Winter chip' the end result would have been truly outstanding. The sound however is a different story. Using sound-sampling EA has managed to use sounds from a real game, and everything from the ball's bounce to the popcorn seller is fabulous.



▲ Archon



▲ Archon

Adventuring

Perhaps the best known of EA's conversions is Archon. As a 64 game I rated it very highly, since it managed to combine the strategy of chess together with a more exciting 'brade' section. On the Amiga it is the same game only better. The graphics are beautifully defined, the sound is fun, and the gameplay is gripping - what more could you ask for?

Out In The Cold

Artificial! If the first three games are good conversions, then Artificial is what the Amiga is all about. Programmed by Dynamix, it may be the best game on any personal computer. To describe the



Flight Simulator

game in full would take hours, but briefly it is a true 3-D (as opposed to sprite 3-D) battlespace-type game, in which you control a 37-ton missile launching 100 mph tanks! This is the only game I have seen that shows what can be done on the Amiga. It has amazing graphics, great sound, and is immensely playable.

VAMIGA



PLAYTIME

On the packaging of *Arctichos* is a line which I think sums up what this game is all about. It reads: "Where do you sleep when you own a 3D-tan tank? Anywhere you want to!"

Arctichos's mission is set in 2020, and you must infiltrate the Alien's force field, and blow up the main fort, thereby defeating the aliens. Against you are a wide variety of stationary as well as mobile weapons which will do their utmost to make sure you get massacred well before you reach the main fort.

What makes *Arctichos* so good is the attention to detail. The instrument panel alone is amazing, with each of the weapons you have being activated by moving an on-screen hand. Radar too has been implemented, and the mini screen used for displaying the guided missiles is incredible.

Reviewing this game poses one difficulty, other reviewers are there but to use! Screen shots cannot do this game justice, and even the fact that it is quite slow doesn't alter the fact that this game is great.

In On The Act

Although EA is the biggest producer of Amiga software, other American software houses see this computer as their chance to be 'there at the beginning'. Activision managed to release three titles very quickly, yet retained a very high standard.



▲ Borrowed Time

As with most of EA's games, Activision has converted three best-selling Commodore 64 titles: *Hayker*, *Borrowed Time*, and *Mindshadow*.

In 1985, *Hayker* was one of Activision's best-selling games. On the Amiga, it has been upgraded graphically, but the gameplay remains the same. You have broken into a computer and must travel around the world collecting sections of a secret document. What makes this game so good on the Amiga is that the small 'monitor' within the game, which is used to display the locations, shows some incredibly well defined pictures of all the major cities around the world (Tower bridge is especially good).

Mindshadow and *Borrowed Time* are both graphical adventures, and although they have no sound, the graphics are of a very high standard. In *Borrowed Time* you play a 1930s

detective called Sam Harlowe, and it is your job to crack a complicated case, and avoid being murdered - not an easy task!

Mindshadow is a more impressive adventure, in which you play a victim of amnesia who must find out who and where he is. Although neither of these games use the full potential of the Amiga, they were created fast, and as such bode well for what Activision will do in the future.

Although England is well behind on Amiga development, we do have one game - *Bratucci*, from Liverpool's Progress. Originally a C64 game, more recently it has been released for the Atari 5200, Mac and Amiga. It is an odd game, in which you must wander around a spacecraft, doing battle using your sword and rebuking others of drinks at the bar! Unfortunately *Bratucci* is another example of a game that does not take advantage of APL of the Amiga's facilities, more down to the reduced screen size and limited colour.

◀ Another

Another ▶



What Next?

What the future holds for Amiga games is uncertain, but I have seen sneak previews of two games which should be released towards the end of 1986. *Wynd Walker* will be Commodore's first, and quite possibly only, game for the Amiga. It is an arcade adventure with you playing the hero, a wizard. As yet *Wynd Walker* has no gameplay, but enormous potential.

The Amiga is most suited to 3-D simulations, and I have seen an unfinished flight simulator that, when it is released, will have airline pilots agree! Programmed by the team responsible for the now legendary flight simulator II for the IBM PC, it is fast, has wonderful solid objects and great potential. Unfortunately in the version I have, you can also fly UNDER the runway, and THROUGH the Pyramid!

As yet, no one has written a game purely for the Amiga, using all the potential of this incredible machine. However I know for a fact that EA is writing the arcade class *Marble Madness*, which should be out within two to three months. All eyes are on it and another EA title *Return to Atlantis* - the first game not to make use of the Amiga - to show what really can be done.

Complimentary Bus Routines

IOINT MFM — Initialize Input/Output
READST SBIT — Read status word

To transmit data to a device, the accumulator (acc.) is loaded with the device number, and the LISTEN routine is called. The secondary address (channel number) is then stored in the acc., bits five and six are set (result = s.a. + 800) and the routine SECOND is called. Data characters stored in the acc. are then sent over the bus using CROUT, and the whole sequence is terminated with the UNLSEN routine, which sends an EOI (End Or Identify).

Getting a device to send data over the bus is just as easy. The KERNAL talk routines are used in place of their corresponding LISTEN calls, and data is input using ACPTB. Bit six of the status (ACPTB) is set after a READST call can be checked after each ACPTB call to test for an EOI from the drive.

To open a file or directory buffer, bits five, six, seven and eight of the secondary address value are set (result = s.a. + 810), bits six, seven and eight are set (result = s.a. + 800) to close the file/buffer.

Armed with this information, we can now write our machine code routine. Note that because the LISTEN/SECOND sequence is used three times I have set it aside as a complete routine call. In a program full of disk access, this is often a valuable space saving technique.

This covers all the "Memory" commands and their various permutations. As reference only, I have included details of these less well known commands I mentioned earlier. So far I have only encountered

these commands implemented in elaborate disk protection schemes; they tend to suggest exact usage because of their limiting qualities. As in the case of the memory commands, they are executed by sending the function and its parameters along the command channel.

BLOCK-EXECUTE: Reads a specific track and sector into a previously opened buffer and executes the machine code routine at the start of that buffer.

Format: "B-C";CHR\$(channel);CHR\$(drive no.);CHR\$(track);CHR\$(sector)
 E.g. PRINT #15,"B-C";2;10;70

LINE-JUMP: Apart from the two "Ls" commands used to read (L1/L2A) and write (L2/L2B) specific sectors to and from the diskette, and the two "Ls" commands used to set the drive bus speed (L3/L4) and reset the drive (L5/L6), there are six "jump" functions which, when called, execute code at the beginning of buffer #2 (\$500).

LN(JC) — jump to \$500
LN(JD) — jump to \$501
LN(JE) — jump to \$502
LN(JF) — jump to \$503
LN(JG) — jump to \$504
LN(JH) — jump to \$505

The nature of these commands would suggest a "jump-table" set up to perform varied tasks within the 154K memory, easily called up from an external Basic or machine code routine. For example, a routine which reads section 154000 (buffer 0) and sends it at high speed along the bus could possibly be called by:

PRINT #15,"C";CHR\$(H);CHR\$(S)

Although, so far I have referred only to the 64 and 154K, the above kernel calls and communication protocols are analogous to both the 128K with T40 Disc Drive, and the C128 in 64 mode with 1550-T40 disk drives.

SENDIBL BYTE "M-M";\$00,\$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8,\$9,\$A,\$B,\$C,\$D,\$E,\$F,\$00
SENDLMD
CITBL BYTE "M-C";\$00,\$00
CITEND
EXTBL BYTE "M-E";\$00,\$00,\$00
END

FOR LISTEN ; send listen + second addr.
AND INDI ; finish if error flag set

FOR #500
LOCOPY LDA SENDIBL,A ; get "Memory-Write" string
FOR CROUT ; output byte
FOR
CPS#(SENDIBL-CITBL) ; is it done?
FOR LOCOPY

FOR UNLSEN ; send last character, EOI.
and UNLSEN

FOR LISTEN
FOR #500
LOCOPY LDA EXTBL,X
FOR CROUT ; send "Memory-Execute" string

FOR
CPS#(EXTEND-CITBL)
FOR LOCOPY
FOR UNLSEN

FOR LISTEN
FOR #500
LOCOPY LDA CITBL,A ; send "Memory-Read" string
FOR CROUT

FOR
CPS#(CITEND-CITBL)
FOR LOCOPY

FOR UNLSEN
LDA#(DEVNUM) ; command drive to talk
FOR TALK

LDA#(SCADD + \$40) ; set secondary address
FOR TALK

FOR ACPTB ; input byte
FOR PHA ; store it
FOR UNTLK ; command disk to stop talking

FOR ; retrieve byte
FOR CROUT ; print it
AND RTS

LISTEN LDA#(DEVNUM) ; tell disk to listen
FOR LISTEN

FOR READST ; get status
AND #15500000 ; check bit 7 (Device Not Present)
AND INDI ; error flag set

LDA#(SCADD + \$40) ; set secondary address (bit 0 set)
FOR SECOND

LDA#(ERR) ; clear error flag
INDI RTS

UNLSEN = \$116 : TALK = \$1164
 SECOND = \$113 : TALK = \$1166
 CROUT = \$11A6 : ACPTB = \$11A0
 UNLSEN = \$11A6 : UNTLK = \$11A8
 READST = \$11B7 : CROUT = \$11D2

DEVNUM = 0
 SCADD = 15



'CHIP

Eric Doyle takes another
long hard look inside your
Commodore computer.

THE 6500 SERIES CENTRAL PROCESSOR may be the heart of a computer but it is the slave of the operating system. The OS is either a machine code program or, more usually, the program contained in the Basic ROM. This is the language of convenience devised by Microsoft which the computer uses when it is first powered up.

For Convenience

A language of convenience has nothing to do with the scoundrels on a literary wall. It is merely a computer language used where communications cannot be conducted in the speaker's mother tongue. Basic is such a language; a computer speaks in pure binary but we use English as a stylized form of English has been devised to ease the job of programming. Similarly, a computer does not use variables in its operation, it uses definite memory locations.

The use of variables in Basic is so essential that I used to take it for granted that my old Vic 20 knew what I was talking about when I referred to them. It wasn't long before I started to wonder how it responded to and stored all the variable names I came up with.

One possible answer was that every observable variable name had a space allocated by the ROM from the moment I turned the computer on. Of course this is impossible in a mere 128K of memory. To allocate sufficient space to 26 strings (A-Z) would need a reserved space of 4,168 and when names are taken into account there wouldn't be enough room in a C128!

The only answer is that the computer adds to the list of variables as each is encountered when a program runs. This still means that space has to be allotted in RAM but it also means that there is a limit to the number of variables which can be defined. It doesn't take much effort to verify this by trying to dimension a large array like DIM A\$(256,256).

This list is stored in the same area of memory as the Basic program and effectively reduces the space which is available. The more variables you use, the shorter your program must be.

This means that the computer needs to keep track of where the variable space is to be found. How many types of

variable are there? Integer, string and floating point make three but there are also arrays and defined functions.

How do we find where these are stored? Memory locations 46 to 52 hold the secret. (C128 reserves in C128 mode should add two to all the following values):

loc#	loc2	Description
46	46	Start of variables
47	48	Start of arrays
49	50	End of arrays
51	52	String storage
53	54	End of string storage (end of Basic memory)

The locations pointed to by the contents of this tiny block of memory can be found by multiplying the contents of loc46 and 256 and adding the contents of loc47.

Down to Work

Time to experiment. Enter the following short program and run it:

```
10 A%=257
```

We must find the start and end of variable storage to type in the following:

```
PRINT 256*PEEK(46)+PEEK(47),256*PEEK(48)-1
```

The values obtained will vary depending on the model of your Commodore but you will find the difference between the two values to be seven bytes. This is the amount of space allotted to all integer variables.

Now let's see what's in these seven bytes. Normally we'd use a loop to PEEK each location in turn but this means setting up a new variable and possibly affecting these memory pointers. Luckily, when a variable is set up another pointer pair indicates where the variable is stored. These are locations 71 and 72 (73, 74 in 128 mode). Add the following lines to your program.

```
10 DIM BAVE(1) IN C128 MODE
20 PEEK 82,PEEK(71):DIM POK(256,
PEEK(72)) IN C128
30 POK(0,0)=PEEK(71):DIM POK(256,
PEEK(72)) IN C128
40 LO=PEEK(82)+PEEK(82)*256:DIM
BAVE(256) IN C128
50 FOR I=0 TO 4:PRINT PEEK (LO+I):
NEXT I
```

Running the program should give these values: 755,128,1,0,0,0.

You're probably wondering what all this means. Let's examine the first two

figures. What happens if we subtract 128 from both numbers? We get 627 and 0. AH 65 is the ASCII code for the letter A, our variable's name. What if we'd called it AAVE? Could it be that we'd have found 755 in the second byte? Try it and see. Change your program line 10 variable to AAVE.

Surely! It works. So we now know that the first two bytes of a stored integer variable is the ASCII code of the first two characters of its name plus 128.

The next two numbers reveal their identity if we treat them in a similar way as we treated the variable pointers earlier. This time we must multiply the first number by 256 and add the second number. Put your computer down, you can do this one in your head. Oh, I'm sure, the answer is 257. Well, well, this is the value of the variable. Clever isn't it?

What do the zeros mean? Precisely that, nothing whatsoever! They're just padding and you'll see why later.

Complex Integers

Let's try a larger value for AAVE, like 12765. See, your computer can't take it. Now try 12767. It likes that, why?

To answer that we've got to go binary. Each byte consists of eight switches which can either be off or on and each indicates a different number. From left to right these are 128, 64, 32, 16, eight, four, two, one. The number four would be represented by the four switch being turned on and all the others off. Five would need switch one and four on and 260 would mean all switches were on.

We have found that the stored number is held in two bytes. In this case the Wonderful eight eight bytes leads to the existing group: 32768, 32864, 32960, 32968, 33056, 33152, 33248.

Adding all 32 numbers together we get a maximum value of 65535. Subtracting the highest permissible integer, 32767, from this gives 32768, the highest switch value. If this switch is not used for number storage what purpose does it have?

If the switch is on, the integer is a negative value and if it is off, the number is positive.

The proof of the pudding etc. Give AAVE a value of -257. The resulting values are 254 and 255. This gives 83,276. We know the number is negative so 32768 can be subtracted to give the answer 12571. If 32768 is subtracted again the result is -257!

CHAT'

This is a mathematical isolation. What actually happens is that a negative number is represented in a form known as two's complement.

To get a two's complement number we need to look at the binary representation of the positive value 257. As on switch is represented by one and off is shown as zero. From our previous example we can see that 257 consists of 256+1, so we turn on those switches only:

```
0000000100000000
```

Next we turn all on switches off and all off switches on:

```
1111110011111110
```

The final act is to turn the last switch back on:

```
1111110011111111
```

Evaluating this gives the value 65535. Now try for yourself to add the 14 switches into two groups of eight and see what values you get using the 128, 64, 32, 16, eight, four, two, one series in both cases. You should get 256 and 255.

What would happen if the last switch was on as in this case?

```
Number      0010011100001100
Complement 1101100011111011
+
-----
```

We can't turn the last switch on to complete our two's complement so we turn it off and try the next switch. If it's on, it's on position. Turn it off and move along the line until you find a switch in the off position, remembering to turn off any switches you have to pass.

In the example the third switch itself so we need search no further:

```
Two's comp. 1101100011111100
```

To convert the number back to a 'real' value, reverse the process.

```
0010011100001100
+ 1
-----
0010011100001101
+0010011100001100
= 0101001100001000
= 2560
```

Don't forget the minus: -2560

Play with the program given A4's different identities and then trying work out from the dated PEEKed the number you first thought of.

Alternatively, if all this binary is too much for you, work out the value stored and subtract 65536 from the result to give the negative value.

Highly Strung

Flushed with success, we'll now look at strings. Change line 10 to A\$=FIND IT* and run the program again.

This time the first three values are 65, 128, seven. The next two numbers will vary from machine to machine but the last two will both be zero. Ignore the zeros, they're meaningless. This time the first letter of the variable name is directly represented by its ASCII value but the second letter still has 128 added. The next value is interesting: count the number of letters in the string. Ah, six!

Trust the fourth and fifth figures as a two-byte number and use this equation:

```
PEEK CHR$(PEEK(4*len)+PEEK(5*len)+128)
```

Substitute the values from your program for the words 'fourth' and 'fifth'.

It gives an "I". Enter the line again but put 31 between the first bracket and the command PEEK. I think we've found it!

Now that the computer reads the value from the program line, why do locations 31 and 32 point to zero?

Replace line 10 with this:

```
10 A$=FIND IT*35+"YOURSELF" A$
A$+85
```

Run the program and, in the words of the variable, find it yourself.

Functional Storage

Straight in the deep end with you!

```
10 DEF FN AA(A)=PEEK(7)+256*PEEK(7)
20 LCH=FN AA(B)*7
```

Trace lines 30 and 40 and run the program.

This time the first character is ASCII+128 and the second is straight ASCII. The third number plus the fourth multiplied by 256 gives the memory location which follows DEF FN AA(A) in your program. PEEK is to make sure.

In the definition we created a variable A as well as a function variable AA and the location of this variable is given by the next two bytes. As usual ignore the sixth number which is another zero.

Pause for a while to appreciate the cleverness of the program which uses the function definition to find the location of variable A and then backs tracks seven places to get to the function entry. PRINT LCH+7 should give the same value as the fifth figure plus the sixth figure multiplied by 256.

In my next article I will be saving arrays and introducing you in floating point variables.

Format of Integer Variables

Byte Contents

- 1 ASCII value of first character of name + 128
- 2 ASCII value of second character of name + 128
- 3 High byte of integer value
- 4 Low byte of integer value
- 5 Not used
- 6 Not used
- 7 Not used

Integers range from -32768 to +32768

Format of String Variables

Byte Contents

- 1 ASCII value of first character of name
- 2 ASCII value of second character of name + 128
- 3 Number of characters in string
- 4 Low byte of string storage address
- 5 High byte of string storage address
- 6 Not used
- 7 Not used

String variables can be up to 255 characters long

Format of defined functions

Byte Contents

- 1 ASCII value of first character of name + 128
- 2 ASCII value of second character of name
- 3 Low byte of pointer to function's location
- 4 High byte of pointer to function's location
- 5 Low byte of pointer to function's internal variable
- 6 High byte of pointer function's internal variable
- 7 Not used

Size of function is unlimited

GO BANANAS

Suitable for virtually ALL computers with 5¼" drives whether single or double sided. If a single sided drive, then you will be able to get twice the storage as you can flip these over.



Only £8.65 exc VAT
(£9.95 inc VAT per ten-pack)

Specification:-

These bananas are reversible & double sided double density. This means that you can get **TWICE** the storage space with single sided drives, as these disks can be flipped over. They are of course perfectly suitable for double sided disk drives, and single density applications.

Grown exclusively by:-

DISKING

FREEPOST

Liphook

Hants GU30 7ER

Tel: (0428) 732563 (24hr Order hotline)



FOR

POST too- DISKING

FREEPOST Liphook Hants GU30 7ER

Tel: (0428 732563)

Please rush me _____ boxes of bananas @ £9.95 per box (plus P & P at 50p a box) inc VAT

Ten boxes or more - **POST FREE**

Official Government orders welcome

I claim my FREE 'Go Bananas' badge with every ten-pack

All orders despatched in 4 hours

Name _____

Address _____

tel no: _____

I enclose cheque payable to DISKING or please debit my Access/VISA card No: _____

If coupon clipped - **JUST CALL**

ASSEMBLER

Steve Carle brings you an Editor Assembler for your C128.

THIS UTILITY IS INTENDED for use on a Commodore 128 system operating in 128 mode. The program is fairly simple in assembly go, but it could be useful to someone who perhaps cannot afford a more comprehensive package. Both tape and disk are supported as well as a printer.

Before going on to describe the program in more detail, I will give a brief overview.

In 128 mode, the computer maintains two 64K banks of RAM (RAM 0 and RAM 1). There are a couple of common areas in the memory map to allow programs to operate correctly between banks. There are 16 predefined memory configurations. C0 (RAM, BANK 1) is used for this program. This is a combination of RAM 0 from 8400-7FFF hex and the local ROM from 8000 hex upwards. This allows the program to make direct calls to the kernel routines without having to go through a complicated bank switching routine. This area from 0000 to 0FFF hex is a common area in all banks. The source code is edited in BANK 1 from 84000-7FFF hex (approx. 62,700) although the actual amount allocated may be altered.

The assembler will allow code to be assembled in any of the predefined banks (actually only RAM 0 and RAM 1 are valid unless you have a 256K machine), and a limited relocation facility is provided. The Commodore machine language monitor (MLM) can be accessed at any time from the editor's command mode. The MLM 5 command will re-enter the editor.

The program will be given in two sections, both of which I will deal with the editor. When you have typed in the editor you may use any of the commands except C which calls the assembler. Without the assembler code the system will probably crash.

The Editor

First type in and save the small program RELOCATE. This program reverts back to a higher address in RAM 0 so as to allow the assembler to be constructed in the correct place.

Note

You must always run this program before using the LOADER.

Now type in and save the program LOADER. Use this to enter the hex data from the main listing. It will be a long job in my opinion is that you do it in stages. That way you won't get frustrated if you keep making mistakes.

If you look at the main hex dump, you will see that each line has an address, a hex data

string and a checksum value. When you run the LOADER, the menu will appear. Options two and three are used to load the complete program area, the memory occupied by both the editor and the assembler (which will be published in the near future).

When asked for the start address, enter 1000. If you save an incomplete version of the program and intend to resume at a later time, make a note of the next address you have to enter and use this when asked for the start address. When you have entered the whole program, type (END) to return to the menu.

During entry the program uses the checksum to validate the input. It will request re-input if an error is found otherwise it will prompt you with the address of the next data string.

Keep these two programs for use with the assembler.

Running the Editor

The program can be loaded and saved like a basic program. Type RUN to enter the editor. A message is displayed and the computer locked into lower case mode. All commands must be entered in lowercase although uppercase may be used in source text. The assembler translates everything into lowercase anyway. The exception to this is the test directive which will be explained when you get the assembler.

The Editor in Operation

The operation of the editor is similar to that of the normal Commodore line editor. Lines are entered with line numbers and the cursor keys may be used to move around the screen.

When first starting out on the program, I suggest you think carefully about how big the source code is likely to be and allocate as necessary.

Unlike the Commodore editor, this program does not remove spaces (max line length is 255 characters). This means that you can make your text more readable by indenting sections of code.

A list of error messages follows. Most are self explanatory.

Error Messages

- Invalid or badly formatted command.
- Invalid sub-command.
- Line does not exist.
- Invalid or missing parameters.
- END error.
- Invalid/lost or range line number.
- No source program.
- String too long.
- Search fails (just really an error).
- Out of memory (program too big).
- Search string too long.
- Replace string too long.
- Illegal device specification.
- Cannot access device while printer is engaged.
- Not valid command for tape.

The Editor Commands

- H Display help message
- A Auto line numbering on/off
- B Set bottom of text memory in RAM 1
- C Compile
- D Delete block
- F Find file and exit to Basic
- F Display memory allocation
- I I/O
- K Kill program
- L List lines
- M Enter monitor
- O Recover (old) program
- P Printer prefix
- R Remamber lines
- S Search (and replace)
- T Set top of text memory in RAM 1

Editor Commands in Detail

- H — Display help page. This displays a summary of the editor commands.
- A — Auto line numbering. This enables and disables the automatic line numbering during program entry. The format of the command is A <increment> in A <block> which sets an increment of 10. Auto numbering will commence from the last line number entered plus the increment. The operation of this command is similar in most respects to the Basic 7.0 auto command.
- B — Set bottom of text memory in Ram 1. Initially the allocation to the editor in RAM 1 is about 6553. This command along with T alters that allocation. Addressing is done in blocks of 256 bytes, numbered 0 to 255. Giving the command B20 will set the bottom of text to block 20 (actual address is 256*(block+4) therefore this address would be 6144 decimal or 1800 hex). The message ARE YOU SURE is printed and the user must give the Y response before the relocation is carried out. This is done since this command destroys any program in memory.
- C — Compile. Details will be given with the assembler listing.
- D — Block delete. Format is D <start>—<end> as in D 20-250. Deletes a block of lines.
- F — Find edit and exit. The message ARE YOU SURE is printed and the user must give the Y response. This is done since exiting the editor may destroy the program in RAM 1.
- F — Display current text memory allocation and number of bytes free.
- I — Input/output. There are several forms of this command.
 - I — Display I/O information
 - I — Display current device directory
 - ID — set current device
 - IN — Set current filename
 - IS — Save file
 - IL — Load file

An important concept is that of the current device and filename. For example:

```
Enter ID
Editor responds
CURRENT DEVICE =0 (DMS)
ENTER NEW DEVICE>
Enter 1
Editor responds
NEW DEVICE =1 (TAPE) OK.
Now enter IN
Editor responds
CURRENT FILENAME =""
ENTER NEW FILENAME>
Enter "test" including the quotes
Editor responds
NEW FILENAME ="test" OK.
Now enter I
The editor prints the following:
CURRENT DEVICE =1 (TAPE)
CURRENT FILENAME ="TEST"
PRINTER IS OFF
```

When B or IL is used the current device and name are used. Note that ID and IN may be used as follows:

```
ID1
IN"test"
```

The IC command will display the directory of the current device if it is a disk drive.

B and IL always load and save from/to the current base block as determined by the B command. These two commands will fail if no filename has been set or the printer is on.

- K — Kill program. Simply deletes the current file. Confirmation is required. The program may, under certain circumstances, be recovered with the O command.
- L — List lines. Format is L <start>—<end>. Run/stop may be used to halt the listing.
- M — Enter MML Monitor's X command will re-enter editor command mode.
- O — Recover deleted program.
- P — Printer prefix. Prefixing most commands with this will cause output to be deflected from the screen to the printer. Will not work with ILIS and IC.
- R — Remamber lines. Format is R <start>,<increment> as in R 10,20.
- S — Search. There are two different format to this command. 1. S <string1>,"<string2>"<start>,<end> 2. S <string1>,"<start>"<end>,<end>. Form 1 finds every occurrence of <string1> and replaces it with <string2>. If "" is used for <start> or <end>, scanning will be from/to the beginning/end of the file. Form 2 finds every occurrence of <string1>.
- T — Set top of RAM 1. Same type of parameters as B. Note that bottom cannot be greater than top.

PROGRAM: RELOCATE

```
10 BANK0:POKE16384,0
20 POKE 46,64
30 PRINT " (DOWN)NOW RUN "CHR$(34) "
  LOADER"CHR$(34)
40 NEW
```

PROGRAM: LOADER

```

10 DO
20 SCHCLR
30 PRINT "ASEM 128 HEX LOADER"
40 PRINT
50 PRINT "1. ENTER HEX DATA"
60 PRINT "2. SAVE CURRENT WORK FILE"
70 PRINT "3. LOAD CURRENT WORK FILE"
80 PRINT "4. END"
90 PRINT "(DOWN) PLEASE CHOOSE OPTION"
100 DO:GETKEY$=A:VAL(A$):LOOP UNTIL A>0 AND A<5
110 :
120 :
130 ON A GOSUB 1000,2000,3000
140 LOOP UNTIL A=4
150 END
160 :
170 :
1000 REM *****
1010 REM HEX ENTRY ROUTINE
1020 REM *****
1030 :
1040 SCHCLR
1050 :
1060 DO
1070 : INPUT "ENTER ADDRESS (IN HEX) ";S#
1080 LOOP UNTIL S#<>" "
1090 S=DEC(S#)
1100 :
1110 PRINT "NOW ENTER DATA AS IT IS PRINTED IN THE LISTINGS. THE COMPUTER WILL PROMPT YOU WITH THE ADDRESS."
1120 PRINT "YOU SHOULD ENTER THE DATA STRING FOLLOWED BY A COMMA THEN THE CHECKSUM VALUE AT THE END OF EACH LINE."
1130 PRINT "ENTER 'END.0' TO EXIT TO MENU."
1140 :
1150 PRINT
1160 :
1170 DO
1180 : DO
1190 : TT=0:HS="" :CS=""
1200 : PRINT HEX$(S)

```

```

1210 : INPUT HS,C#
1220 : IF HS=""END" THEN EXIT
1230 :
1240 : FOR X=1 TO 64 STEP 2
1250 : AD=(X-1)/2+S
1260 : BY=DEC(MID$(HS,X,2))
1270 : BANK0:POKEAD,BY
1280 : TT=TT+BY
1290 : NEXT
1300 :
1310 : IF TT>DEC(C#) THEN PRINT "DATA ERROR. RE-ENTER THIS LINE"
1320 :
1330 : LOOP UNTIL TT=DEC(C#)
1340 :
1350 : S=S+32
1360 LOOP UNTIL HS=""END"
1370 RETURN
1380 :
1390 :
2000 REM *****
2010 REM SAVE CURRENT WORK FILE
2020 REM *****
2030 :
2040 GOSUB 5000 SETUP
2050 PRINT "SAVING WORK AREA"
2060 POKE 253,1:POKE 254,28
2070 SYS DEC("FFD8"),253,192,82
2080 :
2090 PRINT D$
2100 SLEEP 2
2110 RETURN
2120 :
2130 :
3000 REM *****
3010 REM LOAD CURRENT WORK AREA
3020 REM *****
3030 :
3040 GOSUB 5000 SETUP
3050 PRINT "LOADING WORK AREA"
3060 SYS DEC("FFD8"),0,1,28
3070 :
3080 PRINT D$
3090 SLEEP 2
3100 RETURN
3110 :
3120 :
4000 REM *****

```

```

*****
4010 REM COLLECT DEVICE
4020 REM *****
*****
4030 :
4040 INPUT "DEVICE NUMBER ";D
4050 INPUT "FILENAME ";A$
4060 RETURN
4070 :
5000 REM *****
*****
5010 REM SETUP FOR LOAD/SAVE
5020 REM *****
*****
5030 :
5040 GOSUB 4000
5050 BANK 12
5060 SYS DEC("FFFA"),1,D,0
5070 :
5080 SYS DEC("FFA0"),0,1
5090 :
5100 BANK 1
5110 PT=PTRINT(A$)
5120 L=PEEK(PT):LO=PEEK(PT+1):HI=P
EEK(PT+2)
5130 BANK 12
5140 SYS DEC("FFFD"),L,LO,HI
5150 RETURN

```

PROGRAM: MAIN LISTING

```

1001 10100000F023137389E93F113697000000004CB3234C1324000053003A000000 0582
1002 0000000000000000000000000000000000000000000000000000000000000000 0000
1003 0000000000000000000000000000000000000000000000000000000000000000 04FD
1004 3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A3A 04FD
1005 204041404755414745304444444444F522F434F4050444444445522E0D634F4D40F 08AB
1006 4444F52452031323020544552534944F4E2E0D6130534944C2031303063620732E64 8B1B
1007 3E63E5D6153534540424C4553202F204344444444F5230544552534944F4E2D031 0857
1008 2E300D0000000000000000000000000000000000000000000000000000000000 052D
1009 2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A 06D0
1010 5B45435554464F4E2D0000000000000000000000000000000000000000000000 08F2
1011 09279122FF5B8A62561201C29D02CFA2D632E3E362F2F7A2FA92F04300D415245 09CB
1012 30594F5B520530552453058592FAE293F000D454445544F5310434F4D40414E44 07F5
1013 5338004C20584E5D58204E5D20204C49454530D41205B465D20202020204155 07A6
1014 495D0202052454E554D424552204C49454530D41205B465D20202020204155 07A6
1015 544F204E554D424552494E47204F49492F4F4E5D4245524E5D5D5D5D5D5D5D5D5D 07E7
1016 454C4554452E4C494E45530D3F320202020202020202020202040454C503030544849 08EC
1017 53204C495353414745202045202020202020202020202020202020404940D4920202020 0843
1018 20202020202020492F4F20494E4E4F524D4154494F4E5D49444205B4E5D20202020 072D
1019 205345542049555254554E554249492F4F204445544944305D49E302330535452 0821
1020 3E22205430543043553534554E5449492F4F20444944454E414D455D49532020 07D0
1021 20202020202020202020202020202020202020202020202020202020202020 08D4
1022 30202020204C4F4144204E45572048484C455D49443202020202020202020202020 0893
1023 504041592044495308204449534933544F52590D4949202020202020202020202020 075E
1024 4E495449414C4953453043555353454E5430492F4F204445544944304500482020 0834
1025 20202020202020202048494C4C202044454C45544532020202020202020202020 0742
1026 4C490D4F2020202020202020202020202020202020202020202020202020202020 06E9
1027 534F4753414D00053220C535451494E47382258253232305365453464E473E225D 0800
1028 3A4E5B2C4E5D002E202020202020202020202020202020202020202020202020 084C
1029 414345290D48202020202020202020202020202020202020202020202020202020 0707
1030 54415455530D502020202020202020202020202020202020202020202020202020 0734
1031 43202020202020202020202020202020202020202020202020202020202020 06C3
1032 20202020202020202020454E5330454E4F4E494544F520D421020202020202020 0725
1033 2020345543054455054204D454D4F5259204241515345054302020202020202020 0736
1034 2020345543054455054204D454D4F5259204241515345054302020202020202020 0804
1035 522042414440F520484F334D413544454420434F4045414E44049442541424049 0801
1036 402033542420434F404D414E44044C4E4E4E4E4E4E4E4E4E4E4E4E4E4E4E4E4E4E 0857
1037 5400494E23414C49444204F52204D49533494E4720504252414D455445520049 0860
1038 2F4F204552524F5220494E56414C49442F4F5554204F4562052414E4745204C49 084D

```



```

2981 265521A700A20E20012800A09005A20B4CEE148C3B1CC89620212820F132073A 0850
298A F0034C142920E8B20D0834C1429C92AF0A20132220552396508008A547A44895 0843
29C1 03048102B0223514898A517858920F132F035C0C00C202B23F037AA58348 070E
29E1 A58348E52AF0A2092222055239027B006A8FF85168517A518A81780371C8E38 0C4A
2991 1C68858368858368858368858371CA0888D3B1C8A0A2002C2A024C8E2480090361C 0C2F
29A1 8E051C26232A0A000423998086C0C449F07A200863F1C8E301C9804239908 0841
29B1 08F0F0EE381CEE3F1CC80CEFE3E3F1CA90A420B853D863E2008E361CA201C8C 0DC1
29C1 3A1CB800347821A800813C0900CD0086CC3A1C9F03808CE3D01C1E3E1C8A30 0C7F
29D1 4C3D289C3C1C8E391CD0D34C4F2A8A2088D008B9D00EE8E0490F3A1086CE1C 089F
29FA F059ED04C086D04CE8D0F1A00DA50E853E89400D913DC8C3B1C90F59818853B 0D5E
29C1 8588A53C85857A53D18603A1C8538A0388538A000813D9188F03C8D0F7A83F1C 0E1F
29E1 18690448186903808A8A8386900885858A200103033A2848D000E0F0038D0F8E8 0C2B
29D1 8AA200A4828A4A4A808A48520F631A80889080E2D0C23800490F5890080820 0D46
29A1 5C2220F003C8D0F3A580348A58348201D33AD020EA0C98E203825A9202D3FFA9 0D40
29B1 048A0E2021125A9002D3FF189A20A58348A58348AD020A0C38B203825A9202 0A33
29C1 52FFA904A80C201125A9002D3FF6865486865482AD020A0C371C308A002038C 0D3F
29D1 381CF019A0D02040234820004238583688583A0012D0423F0034C2F29AD3C1C 098E
29FA F0034C132A4084CE24C901F00DC908904C90C908A2804280420F123C9 0C7C
29C1 32801620E822A91EA21C2D0F2AC81889089C1A1CA900918260A2034CEE34A800 08EC
29E1 55828883B13D0C22F018C908F01491828C0D109081A2872CA2038A90801E34C 0E1E
29A1 E234C0D0F0F280AD191C901D801E8A8A0FA8288A8F2A0CA9013088F7FA21A1C 0E97
29B1 E88E88A4218A01C2808F3C00FF805320877FF1D23D07FFAD191C901F003D0 10C9
29D1 412E12AA428AC346004032525C454854204445264493A0003D045485445523D 08B8
29E1 4E4557204445564943453E000243555252454E542046494C454E414D453D2203 07E8
29A1 454E5445523D4845572046494C454E414E453E0234F2E481E3D0384E45572044 0798
29B1 45564843453D004E45572046494C454E414E453D2300D4C4F1444848472046 07C4
29C1 494C452023200D0046324F4D204445564943453D000503A1954948472046494C45 0762
29D1 202200D0544F20444556494345200205D5441504525D00484469334820D050 0737
29FA 52494854485220495320004F4646D0094F4E0C05AD191C8A000203825A8C9D1 07BE
29C1 F007A9FA8281890A8A9FA8284C1125A918A81C301125A9134C23FFA949A80 0D73
29D1 2011254C152C8A4D0A2820112520313C9A004C03FFA90885C3061325AD031D8 08FE
29E1 07A9D0A020C41125A911A020C4112520E82200123D03C8A95A0A2820112520 087B
29A1 232D0F122F0320822A817D01CA51620A82A80191C8A98A02820112520152CA9 0A8C
29B1 95A0282011254C132A420C4CEE3428E822D00D20472CA981A028201125208D23 08FD
29C1 308D3A8A8A0820112520312C4C0A03C2838C28472C28593C4C13A43E882380 094C
29D1 064C012C2C9A4D034C782C2C94E8034CAF2CAE301CF005A30D4AC834C93C089 0D10
29E1 4C1C2D94C0D034C802D2C943D0D34C982E2C949D8034C232F4C1324828235AD1E 0AB3
29A1 1C0065A2034CE324280828A0191CC901D80328E8A8A05A02820112520312CA9 087D
29B1 E4A03820112520152C8A003D3FFA80120703D8A547A448851A8417A91A8A8A4 0C0C
29C1 882D80F20872DA9002D3FF4C132A9A01A20C2868FFA91C3C8A8FFAD1A1CA2 0DAB
29D1 1E8A1C4C8DFF80D1E04C3628AD1E1CD00A2034CEE34A547A44885828A8A001 0C6E
29FA 208423F0052088263AAD191C901D80328C88A987A02820112520312CA94F 0C35
29C1 A02B20112520352C8A0020702D8A08647A448857FF80F8A8A48A02D282 08DB
29D1 7F201D234C1334A547A44885828A03A90A8012D0C234C3828A547A44885828A 0C49
29E1 83A001200423F00A8A02D82E2011254C132A9A01A8FF2D0C23201D23A582188 0A51
29A1 02858A85890085894C13A0D05524F4752414D32853A5534A44A582849D080A 092E
29B1 A80FAE191CA8208A8F8A01A2C2088FFA901A48A82F2888FFA90C0FFA90F28C3 0EAB
29C1 7FF8A031CD025A904AA00208A8FA908288FF2D0C0FFAD1A8A9318D301CA204 0E71
29D1 70C9FF10E8124C342A4C132420C0FFA90420C3FFA908080381C8A8A01AE191C8 0C45
29FA 192FA00208A8FA901A2C2088FFA901A31A8A2F288FFA90C0FFA9048308FFA 072D
29C1 4A42A120C0FF20892F20892F20892F20892F20892F20892F20892F20892F20892F 08F0
29D1 22D0F720C92FF0A8C22F0F720D2FF4CE2A5803D02FF4C8D2E2D0C0FFA9013D 03FF
29A1 C9FF4C13244C3628C0CFF80F48208577F208F0F8A8A028201F8018A018A0204 0F68
29B1 E82A8E191C20192F2041284C1334A948A22FA0C90A8C911CA010A8C321C8D80 08DE
29C1 0A8E01CA4C0C80A9086D08FFA8031CA8E331C8D08A8801CA8A63A52F2611254C 097D
29D1 13240D894449544F522F415353454D434C455238312E3808A038D8E32C90F023 08D5
29FA 209323A917801C320542FC9F8B815CD411CF00280E8D401C80A0A2085A8847 0C94
29C1 4C8C23A934C8E2420E823C90F8F428922A817D0E80242FC9F88086C401C 182B
29D1 90E18D411C188904A82FF8D431C88421C4CE2C21A8164820882A8F8034C13A688 0C26
29E1 54858542053054143453D3D020424C4F434853C0002025854853304652345 07DC
29FA 48580AD411C388D401C18890148A8E1A02F20112548A0A3828A8E2A8E2A8E2A 0A64
29D1 1125AD421C88858A48AD431C8588A88828828A8F7A02F2011254C132420825 0A2F

```

Tony Crowther

**shows how to speed
up the C128 in C64
mode.**

WHEN COMMODORE launched the C128 computer they said that it had a C64 computer inside it. They claimed at the time that this was completely compatible with the normal C64 computer. Well, time has demonstrated that it isn't, as many programs will not work on the C128. Obviously there must be some differences between the normal C64 and the one in the C128.

The differences don't only have to be a bad point. It is possible to use some of them to your advantage if you know what you are doing.

As you probably know the C128 has a PAUSE instruction that blanks the screen and causes it to run in 2MHz mode. You probably didn't know that this last mode is also available from the C64. Below are two example programs that will allow you to use this 'bug' to your benefit. For each program I have supplied an assembly listing for those of you who wish to know how they work. I use the Machine Language assembler, but for those who simply wish to use them I have included a simple BASIC loader that will load the machine code on to your disk or tape. If you are using tapes then change the A1 after the SAVE name in each loader to 1,1.

GO FASTER



Program 1

This machine code program when activated by the ENTER key will access the C128's 2MHz processor so that the instruction speed of the C64 is changed from 1MHz to 2MHz with full screen display. The program doesn't finish there, however. By pressing the 'V' key the screen is blanked out from the bottom up. By blanking more of the screen it is possible to alter the speed of the C64 from 1.2MHz up to 2MHz. The greater the amount of screen blanked the faster the program, the screen can be unblanked by

pressing the 'H' key slowing the C64 down.

Now when this program is running you will limit the ability to talk to any peripherals. Therefore hit RUN STOP/RESTORE is quit. If at this point the screen goes funny then either hit the RESET button or type the following line in blank:

POKE \$128,\$1

RUN STOP/RESTORE does not always reset the 2MHz latch as this is inside the C128 and not the C64, hence the need for this POKE to reset the latch to normal.

Both of the programs in this article can be used on the 128 if you alter the interrupt vectors at \$114 and \$115.

Don't attempt this unless you know what you are doing.

Program 2

This program works in a similar fashion to the above program. However, this time the screen is blanked from the top down, not from the bottom. Blanking in this program is also much quicker than the other version of the program.

So what use are these programs? You could use the routines to improve the speed of calculation programs. Who knows we may even start to get programs that have a faster mode for C128 computers.

PROGRAM: 1 CODE

```

100 00000000 00000000 00000000 00000000
101 00000000 00000000 00000000 00000000
102 00000000 00000000 00000000 00000000
103 00000000 00000000 00000000 00000000
104 00000000 00000000 00000000 00000000
105 00000000 00000000 00000000 00000000
106 00000000 00000000 00000000 00000000
107 00000000 00000000 00000000 00000000
108 00000000 00000000 00000000 00000000
109 00000000 00000000 00000000 00000000
110 00000000 00000000 00000000 00000000
111 00000000 00000000 00000000 00000000
112 00000000 00000000 00000000 00000000
113 00000000 00000000 00000000 00000000
114 00000000 00000000 00000000 00000000
115 00000000 00000000 00000000 00000000
116 00000000 00000000 00000000 00000000
117 00000000 00000000 00000000 00000000
118 00000000 00000000 00000000 00000000
119 00000000 00000000 00000000 00000000
120 00000000 00000000 00000000 00000000
121 00000000 00000000 00000000 00000000
122 00000000 00000000 00000000 00000000
123 00000000 00000000 00000000 00000000
124 00000000 00000000 00000000 00000000
125 00000000 00000000 00000000 00000000
126 00000000 00000000 00000000 00000000
127 00000000 00000000 00000000 00000000
128 00000000 00000000 00000000 00000000
129 00000000 00000000 00000000 00000000
130 00000000 00000000 00000000 00000000
131 00000000 00000000 00000000 00000000
132 00000000 00000000 00000000 00000000
133 00000000 00000000 00000000 00000000
134 00000000 00000000 00000000 00000000
135 00000000 00000000 00000000 00000000
136 00000000 00000000 00000000 00000000
137 00000000 00000000 00000000 00000000
138 00000000 00000000 00000000 00000000
139 00000000 00000000 00000000 00000000
140 00000000 00000000 00000000 00000000
141 00000000 00000000 00000000 00000000
142 00000000 00000000 00000000 00000000
143 00000000 00000000 00000000 00000000
144 00000000 00000000 00000000 00000000
145 00000000 00000000 00000000 00000000
146 00000000 00000000 00000000 00000000
147 00000000 00000000 00000000 00000000
148 00000000 00000000 00000000 00000000
149 00000000 00000000 00000000 00000000
150 00000000 00000000 00000000 00000000
151 00000000 00000000 00000000 00000000
152 00000000 00000000 00000000 00000000
153 00000000 00000000 00000000 00000000
154 00000000 00000000 00000000 00000000
155 00000000 00000000 00000000 00000000
156 00000000 00000000 00000000 00000000
157 00000000 00000000 00000000 00000000
158 00000000 00000000 00000000 00000000
159 00000000 00000000 00000000 00000000
160 00000000 00000000 00000000 00000000
161 00000000 00000000 00000000 00000000
162 00000000 00000000 00000000 00000000
163 00000000 00000000 00000000 00000000
164 00000000 00000000 00000000 00000000
165 00000000 00000000 00000000 00000000
166 00000000 00000000 00000000 00000000
167 00000000 00000000 00000000 00000000
168 00000000 00000000 00000000 00000000
169 00000000 00000000 00000000 00000000
170 00000000 00000000 00000000 00000000
171 00000000 00000000 00000000 00000000
172 00000000 00000000 00000000 00000000
173 00000000 00000000 00000000 00000000
174 00000000 00000000 00000000 00000000
175 00000000 00000000 00000000 00000000
176 00000000 00000000 00000000 00000000
177 00000000 00000000 00000000 00000000
178 00000000 00000000 00000000 00000000
179 00000000 00000000 00000000 00000000
180 00000000 00000000 00000000 00000000
181 00000000 00000000 00000000 00000000
182 00000000 00000000 00000000 00000000
183 00000000 00000000 00000000 00000000
184 00000000 00000000 00000000 00000000
185 00000000 00000000 00000000 00000000
186 00000000 00000000 00000000 00000000
187 00000000 00000000 00000000 00000000
188 00000000 00000000 00000000 00000000
189 00000000 00000000 00000000 00000000
190 00000000 00000000 00000000 00000000
191 00000000 00000000 00000000 00000000
192 00000000 00000000 00000000 00000000
193 00000000 00000000 00000000 00000000
194 00000000 00000000 00000000 00000000
195 00000000 00000000 00000000 00000000
196 00000000 00000000 00000000 00000000
197 00000000 00000000 00000000 00000000
198 00000000 00000000 00000000 00000000
199 00000000 00000000 00000000 00000000
200 00000000 00000000 00000000 00000000
201 00000000 00000000 00000000 00000000
202 00000000 00000000 00000000 00000000
203 00000000 00000000 00000000 00000000
204 00000000 00000000 00000000 00000000
205 00000000 00000000 00000000 00000000
206 00000000 00000000 00000000 00000000
207 00000000 00000000 00000000 00000000
208 00000000 00000000 00000000 00000000
209 00000000 00000000 00000000 00000000
210 00000000 00000000 00000000 00000000
211 00000000 00000000 00000000 00000000
212 00000000 00000000 00000000 00000000
213 00000000 00000000 00000000 00000000
214 00000000 00000000 00000000 00000000
215 00000000 00000000 00000000 00000000
216 00000000 00000000 00000000 00000000
217 00000000 00000000 00000000 00000000
218 00000000 00000000 00000000 00000000
219 00000000 00000000 00000000 00000000
220 00000000 00000000 00000000 00000000
221 00000000 00000000 00000000 00000000
222 00000000 00000000 00000000 00000000
223 00000000 00000000 00000000 00000000
224 00000000 00000000 00000000 00000000
225 00000000 00000000 00000000 00000000
226 00000000 00000000 00000000 00000000
227 00000000 00000000 00000000 00000000
228 00000000 00000000 00000000 00000000
229 00000000 00000000 00000000 00000000
230 00000000 00000000 00000000 00000000
231 00000000 00000000 00000000 00000000
232 00000000 00000000 00000000 00000000
233 00000000 00000000 00000000 00000000
234 00000000 00000000 00000000 00000000
235 00000000 00000000 00000000 00000000
236 00000000 00000000 00000000 00000000
237 00000000 00000000 00000000 00000000
238 00000000 00000000 00000000 00000000
239 00000000 00000000 00000000 00000000
240 00000000 00000000 00000000 00000000
241 00000000 00000000 00000000 00000000
242 00000000 00000000 00000000 00000000
243 00000000 00000000 00000000 00000000
244 00000000 00000000 00000000 00000000
245 00000000 00000000 00000000 00000000
246 00000000 00000000 00000000 00000000
247 00000000 00000000 00000000 00000000
248 00000000 00000000 00000000 00000000
249 00000000 00000000 00000000 00000000
250 00000000 00000000 00000000 00000000
251 00000000 00000000 00000000 00000000
252 00000000 00000000 00000000 00000000
253 00000000 00000000 00000000 00000000
254 00000000 00000000 00000000 00000000
255 00000000 00000000 00000000 00000000
256 00000000 00000000 00000000 00000000
257 00000000 00000000 00000000 00000000
258 00000000 00000000 00000000 00000000
259 00000000 00000000 00000000 00000000
260 00000000 00000000 00000000 00000000
261 00000000 00000000 00000000 00000000
262 00000000 00000000 00000000 00000000
263 00000000 00000000 00000000 00000000
264 00000000 00000000 00000000 00000000
265 00000000 00000000 00000000 00000000
266 00000000 00000000 00000000 00000000
267 00000000 00000000 00000000 00000000
268 00000000 00000000 00000000 00000000
269 00000000 00000000 00000000 00000000
270 00000000 00000000 00000000 00000000
271 00000000 00000000 00000000 00000000
272 00000000 00000000 00000000 00000000
273 00000000 00000000 00000000 00000000
274 00000000 00000000 00000000 00000000
275 00000000 00000000 00000000 00000000
276 00000000 00000000 00000000 00000000
277 00000000 00000000 00000000 00000000
278 00000000 00000000 00000000 00000000
279 00000000 00000000 00000000 00000000
280 00000000 00000000 00000000 00000000
281 00000000 00000000 00000000 00000000
282 00000000 00000000 00000000 00000000
283 00000000 00000000 00000000 00000000
284 00000000 00000000 00000000 00000000
285 00000000 00000000 00000000 00000000
286 00000000 00000000 00000000 00000000
287 00000000 00000000 00000000 00000000
288 00000000 00000000 00000000 00000000
289 00000000 00000000 00000000 00000000
290 00000000 00000000 00000000 00000000
291 00000000 00000000 00000000 00000000
292 00000000 00000000 00000000 00000000
293 00000000 00000000 00000000 00000000
294 00000000 00000000 00000000 00000000
295 00000000 00000000 00000000 00000000
296 00000000 00000000 00000000 00000000
297 00000000 00000000 00000000 00000000
298 00000000 00000000 00000000 00000000
299 00000000 00000000 00000000 00000000
300 00000000 00000000 00000000 00000000
301 00000000 00000000 00000000 00000000
302 00000000 00000000 00000000 00000000
303 00000000 00000000 00000000 00000000
304 00000000 00000000 00000000 00000000
305 00000000 00000000 00000000 00000000
306 00000000 00000000 00000000 00000000
307 00000000 00000000 00000000 00000000
308 00000000 00000000 00000000 00000000
309 00000000 00000000 00000000 00000000
310 00000000 00000000 00000000 00000000
311 00000000 00000000 00000000 00000000
312 00000000 00000000 00000000 00000000
313 00000000 00000000 00000000 00000000
314 00000000 00000000 00000000 00000000
315 00000000 00000000 00000000 00000000
316 00000000 00000000 00000000 00000000
317 00000000 00000000 00000000 00000000
318 00000000 00000000 00000000 00000000
319 00000000 00000000 00000000 00000000
320 00000000 00000000 00000000 00000000
321 00000000 00000000 00000000 00000000
322 00000000 00000000 00000000 00000000
323 00000000 00000000 00000000 00000000
324 00000000 00000000 00000000 00000000
325 00000000 00000000 00000000 00000000
326 00000000 00000000 00000000 00000000
327 00000000 00000000 00000000 00000000
328 00000000 00000000 00000000 00000000
329 00000000 00000000 00000000 00000000
330 00000000 00000000 00000000 00000000
331 00000000 00000000 00000000 00000000
332 00000000 00000000 00000000 00000000
333 00000000 00000000 00000000 00000000
334 00000000 00000000 00000000 00000000
335 00000000 00000000 00000000 00000000
336 00000000 00000000 00000000 00000000
337 00000000 00000000 00000000 00000000
338 00000000 00000000 00000000 00000000
339 00000000 00000000 00000000 00000000
340 00000000 00000000 00000000 00000000
341 00000000 00000000 00000000 00000000
342 00000000 00000000 00000000 00000000
343 00000000 00000000 00000000 00000000
344 00000000 00000000 00000000 00000000
345 00000000 00000000 00000000 00000000
346 00000000 00000000 00000000 00000000
347 00000000 00000000 00000000 00000000
348 00000000 00000000 00000000 00000000
349 00000000 00000000 00000000 00000000
350 00000000 00000000 00000000 00000000
351 00000000 00000000 00000000 00000000
352 00000000 00000000 00000000 00000000
353 00000000 00000000 00000000 00000000
354 00000000 00000000 00000000 00000000
355 00000000 00000000 00000000 00000000
356 00000000 00000000 00000000 00000000
357 00000000 00000000 00000000 00000000
358 00000000 00000000 00000000 00000000
359 00000000 00000000 00000000 00000000
360 00000000 00000000 00000000 00000000
361 00000000 00000000 00000000 00000000
362 00000000 00000000 00000000 00000000
363 00000000 00000000 00000000 00000000
364 00000000 00000000 00000000 00000000
365 00000000 00000000 00000000 00000000
366 00000000 00000000 00000000 00000000
367 00000000 00000000 00000000 00000000
368 00000000 00000000 00000000 00000000
369 00000000 00000000 00000000 00000000
370 00000000 00000000 00000000 00000000
371 00000000 00000000 00000000 00000000
372 00000000 000000
```


Gary Herman brings you the first part of a series which will show you how to make the most of your C64's musical talents.

IT'S A FREQUENTLY REPEATED FACT that Commodore Basic is the worst thing about the 64. From a musical point of view, all the instructions are reduced to POKEs (or PEEKs in some instances) to memory locations corresponding to particular registers on the machine's 6581 programmable sound generator.

This makes for a relatively slow transition to machine-code programming — a topic we'll be dealing with in detail later in the series since sound synthesis and music programming are more effective as you get closer to the hardware level. An introduction to basic techniques is useful because the 64 sound commands are already very close to the hardware level. Unless you use a basic extension, a high-level language or music utility, music programming in Commodore Basic will inevitably bring you to within spitting distance of the computer's hardware. This is not the case with any other popular machine.

For basic use, all you really need to know is that location 54272 corresponds to register zero on the 6581 chip and the locations, like the registers, are numbered in steps of one upwards. This is because the 6581 sound interface device is memory-mapped. It is addressed at location 54000 HEX (54272 DECIMAL). That is, the chip is enabled when address lines A/0, A/4, A/12 and A/16 are all high. The lines A/0 to A/4 are then used to address 512 registers. While there are 512 possible addresses using A/0 to A/4, 512 actually has only 20 registers. The last three addresses (54304, 54308 and 54300 — or, in hex, D410, D414 and D418) are not used.

In general, POKEing a memory location can best be understood as writing certain data lines high and others low. There are three independent sound channels on the Commodore, each one requiring at least five and at most seven different locations to be POKE'd. There are seven locations which relate to all three channels — three write-only locations and four read-only locations. The write-only locations are the 512 registers used to set up the sound you wish to produce. Turning up the sound is just a matter of POKEing the right data into the right location.

Typically, the procedure is first to POKE location 54296 with a volume setting which, as it were, sets up all the channels. This setting is a value between zero (off) and 15 (maximum). Then 54296

can be divided into two nybbles (four bit numbers), the least significant or right-hand nybble comprising bits number three, two, one and zero of the byte addressed at location 54296. POKEing 54296 with, say, nine sets bit three high (1), bit one low (0), bit one low (0) and bit zero high (1). This gives 1001 as our nybble, in

binary code, and 1001 binary is equivalent to nine decimal. The other, high order, nybble at location 54296 is made up of the four most significant bits of the byte — numbers seven, six, five and four. Setting six, five and four high or low has the effect of switching on or off one of the 1000 chip's filter modes. Setting bit seven high or low has the effect of turning off or on the audio output of channel three.

Having set a volume, you must then select the desired channel for output and POKE the two associated locations with

```
100 REM *****
110 REM 100,101100 *****
120 REM 1000000,00 10000 0000 10000
130 REM 100,10000 0000000000000000
140 REM *****
150 S=54272:L=5:R=1:1:R=0:1:R=0:1:R=0:
  A=P=5:G
160 S=5:G=P=54:G=54:G=5:G=1:1:1
170 FOR P=0 TO 20:POKE S+P,G:NEXT
180 POKE R,G:POKE R,G:POKE R,G
190 FOR P=0 TO 10:POKE L+P,POKE V,R:G
  IT
200 POKE R,G
```

READY.

Program Listing 1

READY.

```
200 REM*****
210 REM100,101100 2400000000000000
220 REM100000000 1000000000000000
230 REM*****
240 S=54272:L=5:R=0:1:R=0:1:R=0:1:R=0:
  A=P=5:G
250 S=5:G=P=54:G=54:G=5:G=1:1:1
260 FOR P=0 TO 20:POKE S+P,G:NEXT
270 POKE R,G:POKE R,G:POKE R,G
280 FOR P=0 TO 10:POKE L+P,G:POKE V,R:G
  IT
290 POKE R,G
```

Program Listing 2

```
300 REM *****
310 REM 100,101100 2400000000000000
320 REM 1000000 10000 0000 10000
330 REM *****
340 REM 100,10000 0000000000000000
350 REM *****
360 POKE 54276,15:POKE 54277,0: POKE 5
  4278,15
370 READ A,B,C
380 T=0
390 IF A=0 THEN GOTO
400 POKE 54275,A: POKE 54276,B
410 POKE 54276,C
420 FOR P=0 TO 10:GOTO
430 POKE 54276,P
440 FOR P=0 TO 10:POKE 54276,P:POKE 54276,P
  4400
450 DATA 17,15,20,20,15,20,17,17,20,0
  ,147,15,15,20,15,15,0
```

READY.

DISKING FOR A LISTING A LISTING

Program Listing 3

```

100 REM *****
110 REM *****
120 REM *****
130 REM *****
140 REM *****
150 PRINT "MIDI 1471: 94477: 1-100000.5"
160 REM *****
170 REM *****
180 REM *****
190 REM *****
200 REM *****
210 REM *****
220 REM *****
230 REM *****
240 REM *****
250 REM *****
260 REM *****
270 REM *****
280 REM *****
290 REM *****
300 REM *****
310 REM *****
320 REM *****
330 REM *****
340 REM *****
350 REM *****
360 REM *****
370 REM *****
380 REM *****
390 REM *****
400 REM *****
410 REM *****
420 REM *****
430 REM *****
440 REM *****
450 REM *****
460 REM *****
470 REM *****
480 REM *****
490 REM *****
500 REM *****
510 REM *****
520 REM *****
530 REM *****
540 REM *****
550 REM *****
560 REM *****
570 REM *****
580 REM *****
590 REM *****
600 REM *****
610 REM *****
620 REM *****
630 REM *****
640 REM *****
650 REM *****
660 REM *****
670 REM *****
680 REM *****
690 REM *****
700 REM *****
710 REM *****
720 REM *****
730 REM *****
740 REM *****
750 REM *****
760 REM *****
770 REM *****
780 REM *****
790 REM *****
800 REM *****
810 REM *****
820 REM *****
830 REM *****
840 REM *****
850 REM *****
860 REM *****
870 REM *****
880 REM *****
890 REM *****
900 REM *****
910 REM *****
920 REM *****
930 REM *****
940 REM *****
950 REM *****
960 REM *****
970 REM *****
980 REM *****
990 REM *****

```

bytes which set attack, decay, sustain and release characteristics. The ADNR values must be entered before the rate is actually turned on, which is a feature of the hardware.

ADNR values are represented by *nybbles*—an attack is the high order nybble of one byte and decay the low order nybble of the same byte, while sustain and release are the high and low order nybbles, respectively, of the byte entered into the next location in memory. Thus there are 16 possible values for each of the ADNR parameters, corresponding to the 16 possible values of one nybble. Decay values, for example, are — in decimal — any number between zero and 15, while attack values (as the high order nibble) are any multiple of 16 between 0716 and 15716. They can, of course, be added together to give a combined setting for attack and decay, since we can visualize this process simply as two nibbles bits are set low or high in both nybbles by determining a value for a single byte. If the value in the PDR16 is one and different powers of two, 8, 5, 2, 4, 8, 16, 32, 64, or 128, the effect of adding them is the same as performing a logical OR on them, since PDR16 is a location with one of the above powers of two set previously one data line high. This point should be borne in mind for future reference.

The table below gives the real values corresponding to ADNR parameter settings. Note that A, D and R are given as rates. That is, the lower their value, the faster that phase of the envelope is over; while S is a proportion of peak volume. During the attack phase, the sound rises to the level set by PDR16 54296 (the peak level). During the decay phase, the sound diminishes to a level set by the sustain value (the plateau level). This can be any of 16 values from the peak value (with 1 sustain is set as 154 or zero, setting sustain to eight, for example, would give a plateau level of roughly half the peak value — that is, for all practical purposes, plateau level = peak value \div sustain value/16). All amplitude values on the 84 increase linearly, which is to say that amplitude (which is heard as a linear increase) you need to double the setting; thus an amplitude of four (which is twice an amplitude of two (which is not the case with other computers), Programs 1 and 2 use PDR16 NEXT loops to create decaying notes, the first decreasing in volume or amplitude by linear steps, the second by an approximate halving at each step. Notice the difference in effect. Also note the actual locations used.

Value	Attack	Decay/Release	Sustain
0 (0.0)			
1 (0.0)	0 ms	24 ms	0.00
2 (0.0)	8 ms	48 ms	0.14
3 (0.0)	24 ms	72 ms	0.29

Program Listing 4

given frequency depends on duty cycle. A value of zero POREL into both registers on a value of 4095 (that is, 15 in the high nibble and 255 in the low byte) will give a constant DAC output. A value of 2048 (eight in the high nibble and zero in the low byte) will give a square wave. It is worth experimenting with combinations of waveforms, as above, using different width pulses plus triangle or sawtooth waves, since the logical ANDing involved works on the harmonics, cancelling out some and magnifying others.

Setting the waveform should be the last PORE in any series of commands, because the control registers are also the control registers for any channel. The values for waveforms are as given above plus 128 for noise, but the signal will only be heard if bit zero of the control register is set to one. This is known as the gate bit, and it triggers the start of the attack phase of any sound. If it is set to zero, the sustain phase of the note terminates and it enters its release phase. Thus, to start a note playing, for instance a sawtooth wave, the location corresponding to the relevant waveform/control register must be POREd with 20. To turn the note off, PORE the same location with zero on any even number. Thus a note duration governed by the attack time (given in the table above), plus the proportion of the decay time it takes to reach the sustain level

(given by decay time * (1 — sustain level), where decay time and sustain level are the figures given in the table above for the relevant values. POREd into their associated locations), plus the duration of the sustain phase, plus the release time (given above). Sustain duration is set by means of a delay in the program. This can be almost any command, but typically, a PORE...NEXT loop is used. Other common delay techniques use the 60's internal clock to measure a fixed amount of time (the variable T1 holds clock count, a WAIT command to detect a particular event or a GET command to detect the pressing of a key. Problems sometimes occur with the 60's keyboard buffer, which should be cleared by a PORE 190,0 instruction if a key is pressed to initiate a note.

The remaining sound locations will be dealt with in a future episode of the series, after which we'll move on to machine-code and a discussion of some theoretical aspects of music-making. Meanwhile, you should note that all locations and data used in the sample programs have been ascribed to variables to simplify the actual POREs. This not only makes the programs easier to understand, but it also saves time, since the 60 deals with variables quicker than it does with numeric constants. To finish, Program 6 shows what can be done with the 60's filter facility as a matter of things to come.

SEARCHING FOR A LISTING ALISTON

NEXT.

```

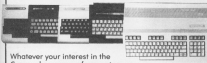
100 REM*****
101 REM***** LISTING 6 *****
102 REM***** FILTER EFFECT*****
103 REM*****
104 POKE 34276,31:POKE 34277,64: POKE
34278,130:POKE 34292,0:PRINT CHR$(147)
150 FOR B=0 TO 240 STEP 240:POKE 34295
,144
160 PRINT CHR$(171);TAB(10);"RESONANCE"
:B
170 READ A,B
180 IF A=0 THEN 210
190 POKE 34276,A: POKE 34278,B
200 POKE 34276,31
210 FOR P=0 TO 255 STEP 5:POKE 34274,P
:NEXT
220 GOTO 170
230 FOR P=1 TO 800: NEXT: POKE 34274,0
240 RESTORE:NEXT
250 DATA 17,21,21,134,17,27,6,147,17,27
64,6,0

```

Program Listing 6

SUBSCRIPTION ORDER FORM

Your COMMODORE



Whatever your interest in the Commodore range of computers and peripherals, you simply cannot afford to miss a single issue.

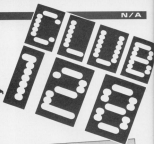
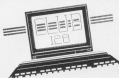
SUBSCRIPTION RATES

£16.60 for 12 issues U.K.
£21.50 for 12 issues Overseas Surface mail
£57.00 for 12 issues Overseas Air mail



Send this form with your remittance to:
INFOMET LTD., Thame House, 179 The Barbican,
Bristol NG1 6JF, Bristol, NPT 168.

Now begin to subscribe to 200 Commodore with the
12 issues in complete time for £1
or £1.50
Full time
NAME (please print)
ADDRESS
Postcode
Signature
Date
Please enclose cheque
or money order



HAVE YOU EVER WANTED TO GET INTO communications? Would you like to get in touch with other Commodore owners? Well, Your Commodore, together with CompuNet, are pleased to announce the start of Club 128, and give you a chance of a special membership offer.

Club 128 will appear on CompuNet as a 'free access' area open to all CMI subscribers.

The club will be the focal point for all 'serious' users of Commodore computers, hence the '128' affiliation. The 128 doesn't mean that owners of C64s can't join. In fact the club is open to all CompuNet members (though only 'serious' users are expected to make a great deal of use of it).

Many sections of CompuNet are due to be 'pulled in' to this online area, including the 'Business' business section that is already in existence.

Your Commodore will have a magazine section within the club area where you will be able to find news and the latest programs that are published in the Magazine.



The Offer

So what do you get and what does it cost?

Your Commodore is offering a special membership to the Club for £89.99 (inc VAT and P&P in the UK) including: A Commodore 64/128 Communications modem (worth £79.99).

A three-month GOLD subscription to CompuNet (worth £15.00).

A six-month subscription to Your Commodore.

* PLUS *

£10.00 of discounts on selected business and communications software, including software for accessing Viewdata and Elextrans Mail services such as Printer and Telecom Gold.

In order to take advantage of this special offer you simply need to send a cheque or Postal Order made payable to Club 128 to the address on the below order form.

I enclose a cheque made payable to Club 128 for £89.99 for my Gold membership to CompuNet, Commodore modem and six month subscription to Your Commodore.

Name

Address

Postcode

Please allow at least 28-days for delivery. Order should be sent to Club 128/CompuNet Telecommunications Ltd, 7-11 Minerva Rd, London NW10 6PL.

Save your fingers a lot of
work with our new software
service.

SOFTWARE FOR SALE

IT'S THREE O'CLOCK IN THE MORNING. You sit at the computer keyboard having just finished a marathon typing session entering one of the superb programs from Your Commodore. Your fingers reach for the keyboard and press the letters R, U and N. You sit back expectantly and... nothing happens.

Well, I'm sure that we have all had problems before now. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

Here, at Your Commodore, we pride ourselves on the quality of listing that we print. Unfortunately, this usually means that they are also very long, thus taking longer to type in and leaving more room for errors. All of the listings in Your Commodore are taken straight from a printed set of working programs, it is therefore very unusual for errors to appear in the magazine.

Because of the length of our programs we also get a large number of requests from readers who would like us to put specific

programs on tape or disk for them. Obviously this is very time-consuming and means that we can't spend as much time working on the magazine as we would like.

We are therefore proud to announce the start of the 'Your Commodore Software Service'. Most of the programs from each issue of the magazine will now be available on a single cassette for a price of just £4.00. We will not be making disks available since they would have to be a lot more expensive and more difficult to post. This shouldn't cause you any

problems though as most of the programs will be protected and it will be a simple matter to save the programs to disk yourselves.

All programs on the cassette will be saved using a type turbo routine. However, we cannot guarantee that all programs will work correctly with this turbo routine power. We therefore recommend that before you use any of the programs you make a copy of the programs on your own cassette or disk and use this version of the program not the original.

PLEASE COMPLETE IN BLOCK CAPITALS

All cheques/postal orders should be made payable to:
ARGUS SPECIALIST PUBLICATIONS LTD

ORDER FORM

I would like to order the following cassettes: (please tick box)

APR '86	YCAPR86	£4.00	<input type="checkbox"/>
MAY '86	YCMAY86	£4.00	<input type="checkbox"/>
JUNE '86	YCJUN86	£4.00	<input type="checkbox"/>
JULY '86	YCJUL86	£4.00	<input type="checkbox"/>

I enclose a cheque/postal order for £.....

NAME.....

ADDRESS.....

.....

.....

POSTCODE.....

All orders should be sent to:
YOUR COMMODORE SOFTWARE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS,
WOLSEY HOUSE, WOLSEY ROAD, HEMEL HEMPSTEAD, HERTFORDSHIRE HP3 1SS.

Please allow at least 28 days for delivery.



John Fletcher proves
that good games are
possible in Basic.

YOUR CITY IS UNDER attack from the alien "Horker". Your mission is to rescue all of your people from the city before "Horker" reaches it. Should he reach it, then you can wave good-bye to your three lives.

There may seem to be nothing original about this game but there are a couple of things that make it worth looking at. Firstly it is fun to play. Secondly, John Fletcher has proved that not all good games need to be written totally in machine code. Lunar Ordeal is written nearly in Basic.

A twist: in part 2 is needed to play the game.

Variables

Scoring Variables and strings:
SCORE,SHOTS,CA,ABC,AVG,QL,AV
Module Movement:
W,XY,XY2,XY3,XY4,XY5,LD

Microhouse:
L — lives
L — level

ST — joystick movement

ST — joystick movement

ST — fuel

ST — Print title page if ST=0

PROGRAM: LUNAR ORDEAL

```

1 DIM SH=100,SC=100
2 FOR T=0 TO 10
3   SH=ST+700 SH VET 10
4   SC=ST+500 SC=ST+VET
5 IF SH=0 THEN GOTO 1000
6 IF SC=0 THEN GOTO 1000
7 IF SH=0 THEN GOTO 1000
8 IF SC=0 THEN GOTO 1000
9 IF SH=0 THEN GOTO 1000
10 IF SC=0 THEN GOTO 1000
11 IF SH=0 THEN GOTO 1000
12 IF SC=0 THEN GOTO 1000
13 IF SH=0 THEN GOTO 1000
14 IF SC=0 THEN GOTO 1000
15 IF SH=0 THEN GOTO 1000
16 IF SC=0 THEN GOTO 1000
17 IF SH=0 THEN GOTO 1000
18 IF SC=0 THEN GOTO 1000
19 IF SH=0 THEN GOTO 1000
20 IF SC=0 THEN GOTO 1000
21 IF SH=0 THEN GOTO 1000
22 IF SC=0 THEN GOTO 1000
23 IF SH=0 THEN GOTO 1000
24 IF SC=0 THEN GOTO 1000
25 IF SH=0 THEN GOTO 1000
26 IF SC=0 THEN GOTO 1000
27 IF SH=0 THEN GOTO 1000
28 IF SC=0 THEN GOTO 1000
29 IF SH=0 THEN GOTO 1000
30 IF SC=0 THEN GOTO 1000
31 IF SH=0 THEN GOTO 1000
32 IF SC=0 THEN GOTO 1000
33 IF SH=0 THEN GOTO 1000
34 IF SC=0 THEN GOTO 1000
35 IF SH=0 THEN GOTO 1000
36 IF SC=0 THEN GOTO 1000
37 IF SH=0 THEN GOTO 1000
38 IF SC=0 THEN GOTO 1000
39 IF SH=0 THEN GOTO 1000
40 IF SC=0 THEN GOTO 1000
41 IF SH=0 THEN GOTO 1000
42 IF SC=0 THEN GOTO 1000
43 IF SH=0 THEN GOTO 1000
44 IF SC=0 THEN GOTO 1000
45 IF SH=0 THEN GOTO 1000
46 IF SC=0 THEN GOTO 1000
47 IF SH=0 THEN GOTO 1000
48 IF SC=0 THEN GOTO 1000
49 IF SH=0 THEN GOTO 1000
50 IF SC=0 THEN GOTO 1000
51 IF SH=0 THEN GOTO 1000
52 IF SC=0 THEN GOTO 1000
53 IF SH=0 THEN GOTO 1000
54 IF SC=0 THEN GOTO 1000
55 IF SH=0 THEN GOTO 1000
56 IF SC=0 THEN GOTO 1000
57 IF SH=0 THEN GOTO 1000
58 IF SC=0 THEN GOTO 1000
59 IF SH=0 THEN GOTO 1000
60 IF SC=0 THEN GOTO 1000
61 IF SH=0 THEN GOTO 1000
62 IF SC=0 THEN GOTO 1000
63 IF SH=0 THEN GOTO 1000
64 IF SC=0 THEN GOTO 1000
65 IF SH=0 THEN GOTO 1000
66 IF SC=0 THEN GOTO 1000
67 IF SH=0 THEN GOTO 1000
68 IF SC=0 THEN GOTO 1000
69 IF SH=0 THEN GOTO 1000
70 IF SC=0 THEN GOTO 1000
71 IF SH=0 THEN GOTO 1000
72 IF SC=0 THEN GOTO 1000
73 IF SH=0 THEN GOTO 1000
74 IF SC=0 THEN GOTO 1000
75 IF SH=0 THEN GOTO 1000
76 IF SC=0 THEN GOTO 1000
77 IF SH=0 THEN GOTO 1000
78 IF SC=0 THEN GOTO 1000
79 IF SH=0 THEN GOTO 1000
80 IF SC=0 THEN GOTO 1000
81 IF SH=0 THEN GOTO 1000
82 IF SC=0 THEN GOTO 1000
83 IF SH=0 THEN GOTO 1000
84 IF SC=0 THEN GOTO 1000
85 IF SH=0 THEN GOTO 1000
86 IF SC=0 THEN GOTO 1000
87 IF SH=0 THEN GOTO 1000
88 IF SC=0 THEN GOTO 1000
89 IF SH=0 THEN GOTO 1000
90 IF SC=0 THEN GOTO 1000
91 IF SH=0 THEN GOTO 1000
92 IF SC=0 THEN GOTO 1000
93 IF SH=0 THEN GOTO 1000
94 IF SC=0 THEN GOTO 1000
95 IF SH=0 THEN GOTO 1000
96 IF SC=0 THEN GOTO 1000
97 IF SH=0 THEN GOTO 1000
98 IF SC=0 THEN GOTO 1000
99 IF SH=0 THEN GOTO 1000
100 IF SC=0 THEN GOTO 1000

```

LUNAR ORDEAL

```

11 PRINT "HOME,DOWN,GO"
12 #####
13 #####
14 #####
15 #####
16 #####
17 #####
18 #####
19 #####
20 #####
21 #####
22 #####
23 #####
24 #####
25 #####
26 #####
27 #####
28 #####
29 #####
30 #####
31 #####
32 #####
33 #####
34 #####
35 #####
36 #####
37 #####
38 #####
39 #####
40 #####
41 #####
42 #####
43 #####
44 #####
45 #####
46 #####
47 #####
48 #####
49 #####
50 #####
51 #####
52 #####
53 #####
54 #####
55 #####
56 #####
57 #####
58 #####
59 #####
60 #####
61 #####
62 #####
63 #####
64 #####
65 #####
66 #####
67 #####
68 #####
69 #####
70 #####
71 #####
72 #####
73 #####
74 #####
75 #####
76 #####
77 #####
78 #####
79 #####
80 #####
81 #####
82 #####
83 #####
84 #####
85 #####
86 #####
87 #####
88 #####
89 #####
90 #####
91 #####
92 #####
93 #####
94 #####
95 #####
96 #####
97 #####
98 #####
99 #####
100 #####

```

```

11 PRINT "DOWN,WHITE,SCORE"
12 "DOWN,WHITE,SCORE"
13 "DOWN,WHITE,SCORE"
14 "DOWN,WHITE,SCORE"
15 "DOWN,WHITE,SCORE"
16 "DOWN,WHITE,SCORE"
17 "DOWN,WHITE,SCORE"
18 "DOWN,WHITE,SCORE"
19 "DOWN,WHITE,SCORE"
20 "DOWN,WHITE,SCORE"
21 "DOWN,WHITE,SCORE"
22 "DOWN,WHITE,SCORE"
23 "DOWN,WHITE,SCORE"
24 "DOWN,WHITE,SCORE"
25 "DOWN,WHITE,SCORE"
26 "DOWN,WHITE,SCORE"
27 "DOWN,WHITE,SCORE"
28 "DOWN,WHITE,SCORE"
29 "DOWN,WHITE,SCORE"
30 "DOWN,WHITE,SCORE"
31 "DOWN,WHITE,SCORE"
32 "DOWN,WHITE,SCORE"
33 "DOWN,WHITE,SCORE"
34 "DOWN,WHITE,SCORE"
35 "DOWN,WHITE,SCORE"
36 "DOWN,WHITE,SCORE"
37 "DOWN,WHITE,SCORE"
38 "DOWN,WHITE,SCORE"
39 "DOWN,WHITE,SCORE"
40 "DOWN,WHITE,SCORE"
41 "DOWN,WHITE,SCORE"
42 "DOWN,WHITE,SCORE"
43 "DOWN,WHITE,SCORE"
44 "DOWN,WHITE,SCORE"
45 "DOWN,WHITE,SCORE"
46 "DOWN,WHITE,SCORE"
47 "DOWN,WHITE,SCORE"
48 "DOWN,WHITE,SCORE"
49 "DOWN,WHITE,SCORE"
50 "DOWN,WHITE,SCORE"
51 "DOWN,WHITE,SCORE"
52 "DOWN,WHITE,SCORE"
53 "DOWN,WHITE,SCORE"
54 "DOWN,WHITE,SCORE"
55 "DOWN,WHITE,SCORE"
56 "DOWN,WHITE,SCORE"
57 "DOWN,WHITE,SCORE"
58 "DOWN,WHITE,SCORE"
59 "DOWN,WHITE,SCORE"
60 "DOWN,WHITE,SCORE"
61 "DOWN,WHITE,SCORE"
62 "DOWN,WHITE,SCORE"
63 "DOWN,WHITE,SCORE"
64 "DOWN,WHITE,SCORE"
65 "DOWN,WHITE,SCORE"
66 "DOWN,WHITE,SCORE"
67 "DOWN,WHITE,SCORE"
68 "DOWN,WHITE,SCORE"
69 "DOWN,WHITE,SCORE"
70 "DOWN,WHITE,SCORE"
71 "DOWN,WHITE,SCORE"
72 "DOWN,WHITE,SCORE"
73 "DOWN,WHITE,SCORE"
74 "DOWN,WHITE,SCORE"
75 "DOWN,WHITE,SCORE"
76 "DOWN,WHITE,SCORE"
77 "DOWN,WHITE,SCORE"
78 "DOWN,WHITE,SCORE"
79 "DOWN,WHITE,SCORE"
80 "DOWN,WHITE,SCORE"
81 "DOWN,WHITE,SCORE"
82 "DOWN,WHITE,SCORE"
83 "DOWN,WHITE,SCORE"
84 "DOWN,WHITE,SCORE"
85 "DOWN,WHITE,SCORE"
86 "DOWN,WHITE,SCORE"
87 "DOWN,WHITE,SCORE"
88 "DOWN,WHITE,SCORE"
89 "DOWN,WHITE,SCORE"
90 "DOWN,WHITE,SCORE"
91 "DOWN,WHITE,SCORE"
92 "DOWN,WHITE,SCORE"
93 "DOWN,WHITE,SCORE"
94 "DOWN,WHITE,SCORE"
95 "DOWN,WHITE,SCORE"
96 "DOWN,WHITE,SCORE"
97 "DOWN,WHITE,SCORE"
98 "DOWN,WHITE,SCORE"
99 "DOWN,WHITE,SCORE"
100 "DOWN,WHITE,SCORE"

```

```

1100 PRINT "DOWN,SCORE"
1101 "DOWN,SCORE"
1102 "DOWN,SCORE"
1103 "DOWN,SCORE"
1104 "DOWN,SCORE"
1105 "DOWN,SCORE"
1106 "DOWN,SCORE"
1107 "DOWN,SCORE"
1108 "DOWN,SCORE"
1109 "DOWN,SCORE"
1110 "DOWN,SCORE"
1111 "DOWN,SCORE"
1112 "DOWN,SCORE"
1113 "DOWN,SCORE"
1114 "DOWN,SCORE"
1115 "DOWN,SCORE"
1116 "DOWN,SCORE"
1117 "DOWN,SCORE"
1118 "DOWN,SCORE"
1119 "DOWN,SCORE"
1120 "DOWN,SCORE"
1121 "DOWN,SCORE"
1122 "DOWN,SCORE"
1123 "DOWN,SCORE"
1124 "DOWN,SCORE"
1125 "DOWN,SCORE"
1126 "DOWN,SCORE"
1127 "DOWN,SCORE"
1128 "DOWN,SCORE"
1129 "DOWN,SCORE"
1130 "DOWN,SCORE"
1131 "DOWN,SCORE"
1132 "DOWN,SCORE"
1133 "DOWN,SCORE"
1134 "DOWN,SCORE"
1135 "DOWN,SCORE"
1136 "DOWN,SCORE"
1137 "DOWN,SCORE"
1138 "DOWN,SCORE"
1139 "DOWN,SCORE"
1140 "DOWN,SCORE"
1141 "DOWN,SCORE"
1142 "DOWN,SCORE"
1143 "DOWN,SCORE"
1144 "DOWN,SCORE"
1145 "DOWN,SCORE"
1146 "DOWN,SCORE"
1147 "DOWN,SCORE"
1148 "DOWN,SCORE"
1149 "DOWN,SCORE"
1150 "DOWN,SCORE"
1151 "DOWN,SCORE"
1152 "DOWN,SCORE"
1153 "DOWN,SCORE"
1154 "DOWN,SCORE"
1155 "DOWN,SCORE"
1156 "DOWN,SCORE"
1157 "DOWN,SCORE"
1158 "DOWN,SCORE"
1159 "DOWN,SCORE"
1160 "DOWN,SCORE"
1161 "DOWN,SCORE"
1162 "DOWN,SCORE"
1163 "DOWN,SCORE"
1164 "DOWN,SCORE"
1165 "DOWN,SCORE"
1166 "DOWN,SCORE"
1167 "DOWN,SCORE"
1168 "DOWN,SCORE"
1169 "DOWN,SCORE"
1170 "DOWN,SCORE"
1171 "DOWN,SCORE"
1172 "DOWN,SCORE"
1173 "DOWN,SCORE"
1174 "DOWN,SCORE"
1175 "DOWN,SCORE"
1176 "DOWN,SCORE"
1177 "DOWN,SCORE"
1178 "DOWN,SCORE"
1179 "DOWN,SCORE"
1180 "DOWN,SCORE"
1181 "DOWN,SCORE"
1182 "DOWN,SCORE"
1183 "DOWN,SCORE"
1184 "DOWN,SCORE"
1185 "DOWN,SCORE"
1186 "DOWN,SCORE"
1187 "DOWN,SCORE"
1188 "DOWN,SCORE"
1189 "DOWN,SCORE"
1190 "DOWN,SCORE"
1191 "DOWN,SCORE"
1192 "DOWN,SCORE"
1193 "DOWN,SCORE"
1194 "DOWN,SCORE"
1195 "DOWN,SCORE"
1196 "DOWN,SCORE"
1197 "DOWN,SCORE"
1198 "DOWN,SCORE"
1199 "DOWN,SCORE"
1200 "DOWN,SCORE"

```

Program Breakdown

```

0 - Variables for high score table
1 - Check if machine code is in memory
2-3 - Go to main page routine
10-10 - Print scores, set up sprites and variables
110-120 - Main loop for game
1000-1005 - Routine for landing and scoring
1010-1041 - Animate people going into life
1020-1030 - Bring rocket ship down to collect results
1000-2000 - Routine for hitting something, deduct life
1000-3000 - Iterator has reached people, end all games
1000-5000 - Game over, check for high score
5040-5050 - Input name for high score, calculate place,
print names
1000-1060 - Title page
1000-10111 - Read data into appropriate memory
locations, set sprite speeds and directions.
17000-17000 - Data for machine code
20000-20040 - Data for sprites
20000-20090 - Data for user defined members.

```

[illegible][illegible]

[illegible]

```

00000 POINT 'LEFT' = LEFT
00001 POINTTHRU00POINT
00002 POINT 'RIGHT' = RIGHT
00003 POINTSPC1
00004 POINT TAB11BLACK,
00005 DOWNPRESS AND GO TO PL
00006
00007 POINT1SPC,DOWN,BLUE
00008 110SPC,RED11HONKER11
00009 11110SPC11011
00010 111111111111111111
00011 111111111111111111
00012 111111111111111111
00013 111111111111111111
00014 111111111111111111
00015 111111111111111111
00016 111111111111111111
00017 111111111111111111
00018 111111111111111111
00019 111111111111111111
00020 111111111111111111
00021 111111111111111111
00022 111111111111111111
00023 111111111111111111
00024 111111111111111111
00025 111111111111111111
00026 111111111111111111
00027 111111111111111111
00028 111111111111111111
00029 111111111111111111
00030 111111111111111111
00031 111111111111111111
00032 111111111111111111
00033 111111111111111111
00034 111111111111111111
00035 111111111111111111
00036 111111111111111111
00037 111111111111111111
00038 111111111111111111
00039 111111111111111111
00040 111111111111111111
00041 111111111111111111
00042 111111111111111111
00043 111111111111111111
00044 111111111111111111
00045 111111111111111111
00046 111111111111111111
00047 111111111111111111
00048 111111111111111111
00049 111111111111111111
00050 111111111111111111
00051 111111111111111111
00052 111111111111111111
00053 111111111111111111
00054 111111111111111111
00055 111111111111111111
00056 111111111111111111
00057 111111111111111111
00058 111111111111111111
00059 111111111111111111
00060 111111111111111111
00061 111111111111111111
00062 111111111111111111
00063 111111111111111111
00064 111111111111111111
00065 111111111111111111
00066 111111111111111111
00067 111111111111111111
00068 111111111111111111
00069 111111111111111111
00070 111111111111111111
00071 111111111111111111
00072 111111111111111111
00073 111111111111111111
00074 111111111111111111
00075 111111111111111111
00076 111111111111111111
00077 111111111111111111
00078 111111111111111111
00079 111111111111111111
00080 111111111111111111
00081 111111111111111111
00082 111111111111111111
00083 111111111111111111
00084 111111111111111111
00085 111111111111111111
00086 111111111111111111
00087 111111111111111111
00088 111111111111111111
00089 111111111111111111
00090 111111111111111111
00091 111111111111111111
00092 111111111111111111
00093 111111111111111111
00094 111111111111111111
00095 111111111111111111
00096 111111111111111111
00097 111111111111111111
00098 111111111111111111
00099 111111111111111111
00100 111111111111111111

```

[illegible][illegible]

Stuart Cooke takes a look at a package that makes games design easier.

AFTER PLAYING YOUR THREE THOUSANDTH game of 'Beat the Alien' you'll probably think that you can do more with your C64 than just play games. Who knows you may even get the urge to write your own games to occupy yourself! However, there is one very big problem - programming. If you've ever tried to move an object around your C64's screen you will have found that it's not quite as easy as the professional programmers make it look. What you need is some sort of tool that takes the hard work out of games design. Well, Amivision has come to your rescue with Game Maker.

Game Maker is actually a number of tools, each one is aimed at helping you to design and write your games program. The scene maker lets you draw the background for your game. Sprite maker allows you to design the characters that will appear on your screen. Music maker and sound maker allow you to add music and sound effects to your program. The Editor allows you to 'write' a program that links all the above parts together to form a game.

Write is probably not quite the correct word for the way in which you write programs. You very rarely need to actually type anything on the keyboard as all of the programming commands are selected from the screen with a joystick. If the commands need any parameters then these are also selected with a joystick. For example if sprite number one was a dog then you would select the 'sprite 1 is' command. The computer would then get the catalog of available sprites from your disk drive, selecting the dog sprite is simply a matter of moving the joystick until the word 'DOG' appears in the window. You have now told the computer that 'sprite 1 is DOG'. Simple isn't it?

Some people may say that if you are going to have to program the game anyway why bother using a program such as this? Why not just write your program in Basic or machine code instead? If we take a look at the example program that you build up over the first five pages of the manual the reason for using a program like this should become apparent.

Let's assume that we have used the different 'Maker' programs to create a dog sprite, a jungle background and a piece of music. In this case the following code would be used to display the background on the screen. The instruction:

```
SCENE 1 IS BUNGEE 2
```

is used to set the BUNGEE2 to the screen used to save the scene on the disk. Now let's add the DOG sprite at position 40,154 (this is the X and Y co-ordinates on the screen, first we need to select the sprite and then position it. This is done with:

```
SPRITE 1 IS DOG  
SPRITE 1 IS POSITION = 40  
SPRITE 1 IS POSITION = 154
```

Simple isn't it? Now let's animate the dog and move it across the screen. The

following commands do this:

```
SPRITE 1 ANIMATION SPEED = 008  
SPRITE 1 DIR = 004 RIGHT  
SPRITE 1 MOVEMENT SPEED = 004
```

And that's our program finished. Simple isn't it? We also BUMP the program a dog will run across our jungle scene from left to right.

The reason for using a program like this should now be apparent, it is so simple to use.

Obviously there are bound to be limitations with a program like this. The author of Game Maker can hardly be expected to cater for every little effect that every programmer will need. Some of the major limitations are that firstly you can only have two screens to your game and secondly, you can only have eight sprites on the screen at once. If you were writing the program in Basic or machine code you could have as many screens as the computer's memory will allow. You can also get more than eight sprites on the screen at once through careful programming.

Even so it is possible to write a wide variety of 'games' with this program. Games provided on the disk with the program range from 'Chopper', a shoot'em up game to an animated Christmas card, complete with clockwork soldier and Jack in the Box.



The Editors

Obviously the quality of the games that you design with this program depends very much on how good the various Editors are. Well you will be pleased to know that they are all excellent. In fact some of them are better than some stand alone programs that are available. Because the programs are so powerful it is probably worth dealing with each one in turn.

Scene Maker

This is the program that allows you to design your back drops. If you have ever



used a graphics program then you will recognise most of the available commands. This allows you to sketch on the screen in one of the available colours. You can have four different colours on the screen at any one time. All drawings are carried out via the joystick. The functions of Line, Box and Circle are quite obvious and Fill allows you to colour in areas of the screen. It is possible to copy areas of the screen from one position to another as well as Zoom in on a specific area so that it is easier to add fine detail.

One interesting thing about this program is the way that the menu of commands covers the top half of the



screen. The bottom half of the screen is used for drawing. My first thought on trying to draw my own background was how on earth do you draw on the top half of the screen? Then I realised that the Move command scrolls the bottom window so that you can see any area of the background screen. If you wish to see the whole picture, the View command turns off the menu.

Sprite Maker is extremely well designed and very simple to use.

Sprite Maker

As you are no doubt aware, probably the most important part of any game is the sprites. These are the objects that move around the screen, shooting at you, killing you and generally getting in your way. Obviously a good game therefore depends on good quality sprites which are clear and well animated. The Sprite Maker program is extremely powerful, in fact I would go as far to say that it is one of the best sprite editors I have come across and it would stand up well on its own.

Sprite maker has provided for either multi-coloured or single colour sprites. Horizontal and vertical magnification is provided and up to four sprites can be placed edge to edge to create one large object that can be easily manipulated.

Once you have defined your basic sprite you can then go on to alter it slightly and store these new 'pictures' on a different frame. You can then run through the frames, like a piece of film, and make your masterpiece an animation.

All of the details about the size of the sprite and the number of animation frames that it has are stored with the sprite when you save the sprite to disk.

Making Music

The music maker allows you to compose your own tunes for inclusion in your games. Music is entered under joystick control so the whole is entered in a musical shorthand. This consists of three musical staves on which you can place the notes for one of the three available voices. However, different instruments are provided ranging from flute to keyboard and any voice can play any instrument.

Entering music is very simple. Select the voice that you wish to use. Select the length of the note and then place the note in its position on the music sheet.

If you have no musical knowledge at all then my feeling is that you may have some problems entering your own tunes. You could always try the 'look it and see' approach and try moving notes around and listening to what results you



get. Thankfully there are a fair range of tunes supplied on the Game Maker disk ranging from William Tell to Happy Birthday so these should always be something to suit your game.

Sound Maker is the part of the package that lets you play around with a 'real' synthesiser. Again the program is under joystick control and everything is easy to alter. You can alter the type of wave form that you are using, alter the attack, decay, sustain and release of the volume and put the sound through filters. If you don't know what any of the above parameters are then I suggest that you just play around, inside the boxes and see what comes out. It is possible to link together a number of sound to create one effect. For example one sound may be a falling tone, another may be an explosion. Switch them together and you've suddenly got bombs falling from your aircraft.

And on it Goes

As I have previously said, this package does have its limitations and obviously you can't expect to get programs of really high quality out of it. However, it is a very good step up the ladder of designing your own games. Even if you can't program, the package will allow you to do up those masterpieces that you can't have been unable to do anything about. Now only does this package let you try ideas out but it will also allow you to grasp what exactly goes into making up a game so that when you eventually go it on your own and try to write your own programs in either Basic or machine code you'll know exactly what you must do.

Game Maker is an extremely well thought out and easy to use package. The documentation is simple enough to that a fairly new computer owner could be writing games in a few hours. All that is needed is a good imagination.

If you do have some ideas for games then this is a package that you should have in your collection.

WELCOME TO THE MACHINE

16 bit numbers are the

subject of Allen Webb's

essay into machine code.

ARE YOU READY FOR THE DELIGHTS of 16 bit arithmetic. Even if you're not, that's what I intend to deal with in this article.

Up to now we've struggled within the constraints of eight bits, which, as you will have realised, can become very messy.

If we work with 16 bit resolution, suddenly everything is simpler. You will recall that to increase a value by one, we can use the INC instruction. Consider Listing 1:

Listing 1

```
10 ASSEMBLE 901
20 RIM *C000
30 RIM INC 900
40 RIM BNE LCOP
50 RIM INC 901
60 RIM LCOP RT5
70 RIM ]
80 FOR =C00000
90 SYS 124000
100 PRINTPRG(901)*256+PRG(900)
110 NEXT
```

This increments a 16 bit number stored in locations 900 and 901. The low byte, 900, is first incremented in line 100. Line 110 checks to see if the low byte has reached zero. If it has, we need to increment the high byte. This will be obvious if you consider what happens as the low byte approaches zero.

Low byte High byte Combined value

254	0	254
255	0	255
0	1	256
1	1	257

The combined value is the low byte plus 256 times the high byte.

To reduce a memory location, you may recall that DEC is used. Listing 2 does this for a 16 bit number:

Listing 2

```
10 AVAILABLE 901
20 RIM *C0000
30 RIM LDA 900
40 RIM BNE LCOP
50 RIM DEC 901
60 RIM LCOP DEC 900
70 RIM RT5
80 RIM ]
90 FOR =C000000
100 SYS 124000
110 PRINTPRG(901)*256+PRG(900)
120 NEXT
```

The principle of this routine is slightly different. The first step is to check whether the high byte needs reducing. This is done in line 70 which checks for a zero value in the low byte. Line 110 decrements the low byte every time.

You will have noted that there is an intimate relationship between the two bytes in a 16 bit number. Before we look at this, consider what happens when you add two numbers in decimal. If the two numbers add to above 10, you carry over. Remember! Consider this sum:

$$\begin{array}{r} 19+ \\ 3 \\ \hline 22 \end{array}$$

The computer works in the same way in that if the result of adding two numbers exceeds 255, you have a carry to tell you about it. This effectively gives you nine bits and the carry flag is used to break the high bit. Addition with carry is performed by the instruction ADC.

Imagine that you wish to add the numbers 7 and 22. In binary these are:

$$\begin{array}{r} 100000111 \quad 7 \\ + 100001110 \quad 22 \\ \hline 100011101 \quad 29 \end{array}$$

The result is less than 255 so we can still use an eight bit number. Consider the addition of 264 and 264:

$$\begin{array}{r} 111100100 \quad 264 \\ + 111100100 \quad 264 \\ \hline 110011110 \quad 528 \end{array}$$

Suddenly, we have a number greater than 255 and the second byte must be used. The number is split as:

$$100000001 \text{ and } 100011110$$

In practical terms, how do we perform this addition? Consider this sequence of instructions:

```
C15
LDA #264
ADC #264
```

The first step is to clear the carry flag — this prepares the way for the carry. The accumulator is then loaded with the number 264. Lastly, 264 is added to the accumulator with the carry flag set as required. The result of this sequence is to leave the accumulator holding 528 (binary 100011110) and the carry flag set. Consider listing 3:

Listing 3

```
10 ASSEMBLE 901
20 RIM *C0000
30 RIM LDA #0
40 RIM STA 901
50 RIM STA 901
60 RIM CLC
70 RIM LDA 900
80 RIM ADC 901
90 RIM STA 901
100 RIM LDA 900
110 RIM ADC #1
120 RIM STA 901
130 RIM LCOP RT5
140 RIM ]
150 INPUL "A,B",A,B
160 PRG(901)*256+PRG(900) RT5
170 SYS 124000
180 PRINTPRG(901)*256+PRG(900)*256
```

This adds two numbers in 900 and 901 and puts the result in the 16 bit number in 900 and 901. Lines 100 to 120 clear the 16 number and lines 130 to 150 add the two numbers. Line 160 stores the low byte of the result and lines 170 to 180 store the carry in the high byte.

Subtraction is a similar process albeit slightly trickier to understand. The relevant instruction is SBC (Subtract with Carry) and again the carry flag is used to convey information on the result of the subtraction. This time, the carry is cleared if the subtraction results in an underflow (negative number). Don't worry about the mechanism of SBC this time. I'll deal with this complication and other mysteries in the future. For now, simply accept that SBC is the reverse of ADC. Consider listing 4.

Listing 4

```

80 ASSEMBLE $00
90 R154 ~+C000
100 R154 $0C
110 R154 LDA #0
120 R154 LDA #0
130 R154 SBC #0
140 R154 STA #0
150 R154 RTS
160 R154 |
170 INPUT "A,B":A,B
180 POKE $00A:POKE $00B
190 SYS 174096
200 PRINT$(#0)

```

The first step, Line 100, is to set the carry flag. The accumulator is then loaded with the first number and Line 130 subtracts the second number. The result is put into location #0. Try moving about with the routine and see what effect it has when you make B larger than A.

Let us consider another example. The first answer to your last month's homework was rather tacky (even at the end of this article, Listing 5 gives a method using 16 bit arithmetic).

Listing 5

```

80 ASSEMBLE $00
90 R154 ~+C000
110 R154 LDA #0
120 R154 STA #0
130 R154 LDA #0
140 R154 STA #C
150 R154 LDA #0
160 R154 STA #0
170 R154 LDA #0
180 R154 LDA #0
190 R154 LDA #0
200 R154 LDA #0
210 R154 LDA #0
220 R154 LDA #0
230 R154 LDA #0
240 R154 LDA #0
250 R154 LDA #0
260 R154 LDA #0
270 R154 LDA #0
280 R154 LDA #0
290 R154 LDA #0
300 R154 LDA #0
310 R154 LDA #0
320 R154 LDA #0
330 R154 LDA #0
340 R154 LDA #0
350 R154 LDA #0
360 R154 LDA #0
370 R154 LDA #0
380 R154 LDA #0
390 R154 LDA #0
400 R154 LDA #0
410 R154 LDA #0
420 R154 LDA #0
430 R154 LDA #0
440 R154 LDA #0
450 R154 LDA #0
460 R154 LDA #0
470 R154 LDA #0
480 R154 LDA #0
490 R154 LDA #0
500 R154 LDA #0
510 R154 LDA #0
520 R154 LDA #0
530 R154 LDA #0
540 R154 LDA #0
550 R154 LDA #0
560 R154 LDA #0
570 R154 LDA #0
580 R154 LDA #0
590 R154 LDA #0
600 R154 LDA #0
610 R154 LDA #0
620 R154 LDA #0
630 R154 LDA #0
640 R154 LDA #0
650 R154 LDA #0
660 R154 LDA #0
670 R154 LDA #0
680 R154 LDA #0
690 R154 LDA #0
700 R154 LDA #0
710 R154 LDA #0
720 R154 LDA #0
730 R154 LDA #0
740 R154 LDA #0
750 R154 LDA #0
760 R154 LDA #0
770 R154 LDA #0
780 R154 LDA #0
790 R154 LDA #0
800 R154 LDA #0
810 R154 LDA #0
820 R154 LDA #0
830 R154 LDA #0
840 R154 LDA #0
850 R154 LDA #0
860 R154 LDA #0
870 R154 LDA #0
880 R154 LDA #0
890 R154 LDA #0
900 R154 LDA #0
910 R154 LDA #0
920 R154 LDA #0
930 R154 LDA #0
940 R154 LDA #0
950 R154 LDA #0
960 R154 LDA #0
970 R154 LDA #0
980 R154 LDA #0
990 R154 LDA #0

```

The key to the routine is the 16 bit number in locations #0 and #C. Instead of carrying the 1 register 16 times this address, we will keep it set to zero and also the base address. Lines 110 to 300 set the address to the start of the screen (\$B000). We then zero the Y register. The main loop puts an asterisk at the currently addressed location (Lines 320 to 340). Lines 350 and 370 increment the base address by one. Lines 380 to 400 compare the base address to \$D7F0 (the last address of the screen) and keep back if it hasn't been reached. Since we're dealing with 16 bits, two comparisons (Lines 380 and 400)

are required. This is clearly a more satisfactory way of working.

Finally, Listings 6 and 7 give routines for the addition and subtraction of two 16 bit numbers. One number is in locations \$00B-\$00F and the other is in \$0C-\$00F. The resulting number is left in locations \$00B-\$00F.

Listing 6

```

80 ASSEMBLE $00
90 R154 ~+C000
110 R154 LDA #0
120 R154 LDA #0
130 R154 LDA #0
140 R154 LDA #0
150 R154 LDA #0
160 R154 LDA #0
170 R154 LDA #0
180 R154 LDA #0
190 R154 LDA #0
200 R154 LDA #0
210 R154 LDA #0
220 R154 LDA #0
230 R154 LDA #0
240 R154 LDA #0
250 R154 LDA #0
260 R154 LDA #0
270 R154 LDA #0
280 R154 LDA #0
290 R154 LDA #0
300 R154 LDA #0
310 R154 LDA #0
320 R154 LDA #0
330 R154 LDA #0
340 R154 LDA #0
350 R154 LDA #0
360 R154 LDA #0
370 R154 LDA #0
380 R154 LDA #0
390 R154 LDA #0
400 R154 LDA #0
410 R154 LDA #0
420 R154 LDA #0
430 R154 LDA #0
440 R154 LDA #0
450 R154 LDA #0
460 R154 LDA #0
470 R154 LDA #0
480 R154 LDA #0
490 R154 LDA #0
500 R154 LDA #0
510 R154 LDA #0
520 R154 LDA #0
530 R154 LDA #0
540 R154 LDA #0
550 R154 LDA #0
560 R154 LDA #0
570 R154 LDA #0
580 R154 LDA #0
590 R154 LDA #0
600 R154 LDA #0
610 R154 LDA #0
620 R154 LDA #0
630 R154 LDA #0
640 R154 LDA #0
650 R154 LDA #0
660 R154 LDA #0
670 R154 LDA #0
680 R154 LDA #0
690 R154 LDA #0
700 R154 LDA #0
710 R154 LDA #0
720 R154 LDA #0
730 R154 LDA #0
740 R154 LDA #0
750 R154 LDA #0
760 R154 LDA #0
770 R154 LDA #0
780 R154 LDA #0
790 R154 LDA #0
800 R154 LDA #0
810 R154 LDA #0
820 R154 LDA #0
830 R154 LDA #0
840 R154 LDA #0
850 R154 LDA #0
860 R154 LDA #0
870 R154 LDA #0
880 R154 LDA #0
890 R154 LDA #0
900 R154 LDA #0
910 R154 LDA #0
920 R154 LDA #0
930 R154 LDA #0
940 R154 LDA #0
950 R154 LDA #0
960 R154 LDA #0
970 R154 LDA #0
980 R154 LDA #0
990 R154 LDA #0

```

Listing 7

```

80 ASSEMBLE $00
90 R154 ~+C000
110 R154 LDA #0
120 R154 LDA #0
130 R154 LDA #0
140 R154 LDA #0
150 R154 LDA #0
160 R154 LDA #0
170 R154 LDA #0
180 R154 LDA #0
190 R154 LDA #0
200 R154 LDA #0
210 R154 LDA #0
220 R154 LDA #0
230 R154 LDA #0
240 R154 LDA #0
250 R154 LDA #0
260 R154 LDA #0
270 R154 LDA #0
280 R154 LDA #0
290 R154 LDA #0
300 R154 LDA #0
310 R154 LDA #0
320 R154 LDA #0
330 R154 LDA #0
340 R154 LDA #0
350 R154 LDA #0
360 R154 LDA #0
370 R154 LDA #0
380 R154 LDA #0
390 R154 LDA #0
400 R154 LDA #0
410 R154 LDA #0
420 R154 LDA #0
430 R154 LDA #0
440 R154 LDA #0
450 R154 LDA #0
460 R154 LDA #0
470 R154 LDA #0
480 R154 LDA #0
490 R154 LDA #0
500 R154 LDA #0
510 R154 LDA #0
520 R154 LDA #0
530 R154 LDA #0
540 R154 LDA #0
550 R154 LDA #0
560 R154 LDA #0
570 R154 LDA #0
580 R154 LDA #0
590 R154 LDA #0
600 R154 LDA #0
610 R154 LDA #0
620 R154 LDA #0
630 R154 LDA #0
640 R154 LDA #0
650 R154 LDA #0
660 R154 LDA #0
670 R154 LDA #0
680 R154 LDA #0
690 R154 LDA #0
700 R154 LDA #0
710 R154 LDA #0
720 R154 LDA #0
730 R154 LDA #0
740 R154 LDA #0
750 R154 LDA #0
760 R154 LDA #0
770 R154 LDA #0
780 R154 LDA #0
790 R154 LDA #0
800 R154 LDA #0
810 R154 LDA #0
820 R154 LDA #0
830 R154 LDA #0
840 R154 LDA #0
850 R154 LDA #0
860 R154 LDA #0
870 R154 LDA #0
880 R154 LDA #0
890 R154 LDA #0
900 R154 LDA #0
910 R154 LDA #0
920 R154 LDA #0
930 R154 LDA #0
940 R154 LDA #0
950 R154 LDA #0
960 R154 LDA #0
970 R154 LDA #0
980 R154 LDA #0
990 R154 LDA #0

```

I now want to briefly discuss an alternative way of manipulating numbers. Consider the binary number seven:

```
%00000111
```

If the bits are shifted left one place with the left-most bit lost and the right-most bit set to zero, we get:

```
%00001110
```

or the number 14. What we have done is multiply the number by two. Similarly, if you shift the bits right one place, we

divide by two. The instructions ASL (Arithmetic Shift Left) and LSR (Logical Shift Right) perform these functions. To multiply a number in location \$00 by two, you simply use:

```
ASL $00
```

to multiply by four, use:

```
ASL $00
```

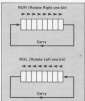
```
ASL $00
```

and so on.

In fact, these instructions do not lose the end bit. As they shift the bit is pushed into the carry flag.



To allow you to make use of the carry flag to manipulate 16 bit or larger numbers, there are a further two instructions:



These rotate the bit pattern, but incorporate the carry bit into the number. Consider the pair of bytes:

```
BYTE 1  BYTE 2
00000000  00000000 = 00
```

Let us shift byte two left once and then roll byte one left once. The left-most bit

MAGIC DISK KIT



Why pay repair bills when you can solve or prevent most of your 1541 and 1570 disk drive problems for only £19.95

Every time you use your disk drive the protection on your software bangs your playing head, eventually knocking it out of place. Now with our very easy to use Magic Disk Kit with a clear book of tips you can remedy head alignment faults and give simply maintenance to your machine to give better performance and longer life.

Available from good computer shops or—

Bobtek

Bobtek Ltd, Unit 4, Isleworth Business Complex,
St John's Road, Isleworth, Middx TW7 8JL,
Telephone 01-847 4457

WELCOME TO THE WORLD of COMMODORE

Maintenance, Service, Support, Training,
from your No 1 **COMMODORE** dealer

Specialist Hardware

HARDWARE

C128D with Monitor & Printer
MPS 1000 NLQ.

SOFTWARE

Accounts (cash book & VAT), W/P,
Database, Spreadsheet.

all for only **£879.00**

Computers

C64 Competition.....189.00
C128 Computer.....265.00
C128D inc disk drive.....409.00
C128D inc d/f, Monitor...585.00
1570 S/S disk drive.....195.00
1571 S/S disk drive.....265.00

Monitors

1901 Colour Monitor.....285.00
1900 80 Column.....145.00
Green/Amber Screens.....95.00
Cable for above.....10.00
C128 40/80 cable.....14.95

Printers

DPS1101 Daisy Wheel..295.00
MPS1000 NLQ inc T/F ..269.00

Software

SuperScript 64...49.00 SuperBase 64...59.00
S/Script 128.....74.00 SuperBase 128 79.00
S/Script + S/Base 64...59.00
S/Script + S/Base 128...139.00

CompuNet Interface 69.99 Disk Interface 69.00

Miracle modem.....115.25 COM modem.....75.00

MS2000 Mouse 69.99

PHONE for FULL LISTS

CHROMASOFT Computer Centres

48 Junction Rd Archway London N10 3BB

288 Maxwell Hill Broadway London N10 3BH

01-288 9406/5 or 01-288 3706

All prices include VAT. Payment by Access,
VISA, Bankers Draft, Building Society Cheques,
Barry cheques need at least 5 days clearance.

Listings will be much easier to enter with our new system.

COMMODORE LISTINGS ARE RATHER well known for the horrible little black blobs that always abound. Unfortunately the graphics characters which are used to represent graphic and control characters do not reproduce very well and they are also difficult to find on the Commodore keyboard.

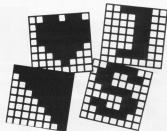
In future all control and graphics commands will be replaced by a mnemonic within square brackets. This mnemonic is not typed out as printed in the magazine but rather the corresponding key or keys on the keyboard are pressed. For example [RIGHT] means press the cursor right key, you do not type in [RIGHT]. All of the keywords, what keys to press and how they are shown on the screen are shown below.

Any character that is accessed by pressing shift and a letter will be printed as (letter).

(A) shift and A
(S) shift and +

Any character that is accessed by pressing the Commodore key and a letter will be printed as (letter).

(CA) Commodore and A
(C+) Commodore and +
(C) Commodore and I



LISTINGS

If any characters are repeated the mnemonic will be followed by a number. This number is how many times you should enter the character. Any number of spaces over one will also be represented in this form.

[RIGHT]10 press cursor right 10 times
[C+]10 press Commodore and + 10 times
[C]10 Press the space bar 10 times

Any other characters should be easily recognisable for example CTRL+N means press CTRL and N and LEFT-ARROW means press the left arrow.

Any number of mnemonics can be enclosed in brackets for example

[SPACE]10[CA]10 means type 10 shift A's 10 spaces and another 10 shift A's.

Mnemonic	Symbol	what to press
[RIGHT]		right
[LEFT]		shift left/right
[UP]		Shift & up/down
[DOWN]		up/down
[F1]		F1
[F2]		shift & F1
[F3]		F3
[F4]		shift & F3

Mnemonic	Symbol	what to press
[F5]		F5
[F6]		shift & F5
[F7]		F7
[F8]		shift & F7
[CLEAR]		shift & CLR /HOME
[HOME]		CLR/HOME
[F9]		CTRL & 9
[F10]		CTRL & 0

Mnemonic	Symbol	what to press
[BLACK]		CTRL & 1
[WHITE]		CTRL & 2
[RED]		CTRL & 3
[CYAN]		CTRL & 4
[PURPLE]		CTRL & 5
[GREEN]		CTRL & 6
[BLUE]		CTRL & 7
[YELLOW]		CTRL & 8

C16/PLUS 4

LEGIONNAIRE

Your mission is to take vital messages to the garrison headquarters. As you travel through the scorching landscapes, you will be under constant fire from the tribesmen bent on reaching the fort. You will have to use your machine gun with great skill to get past them. There are seven separate areas of scorching landscapes before the fort can be reached and each area requires increasing skill.

C16/PLUS 4 £5.95



BONGO CONSTRUCTION SET

Bongo Construction Set is a brilliant family game. Bongo is trying to reach the princess, badly pursued by monsters. He can build, attack, transport, lift and transport to solve his problems but they seem to know his moves before he makes them. There are six stages with five skill tests. You can design your own stages using the joystick and save them on cassette or disk to play later.

C16/PLUS 4 £5.95

SPACE PILOT

Space Pilot, the legendary hero, is under constant attack by alien ships with one aim - the destruction of his craft. The alien craft attack singly or in battle formation. Equipped with laser seeking missiles, their attack is deadly as well as relentless. Space Pilot uses full 360 degree scrolling and fast stages, demanding ever increasing skills testing all aspects of a smooth hit on little bit.

C16/PLUS 4 £5.95

TAIJI BOXING

A unique combination of the noble art of Taiji and skills of Martial arts. Three stages of increasing skill.

16/PLUS 4 £5.95



FORTHCOMING ANCO RELEASES



WATER SPORTS
Race, Sink and Sinking the Rapids.

C16/PLUS 4 £5.95

WATER SPORTS PLUS
Enhanced version for PLUS 4 owners or C16 with 16K expansion includes an extra event, Power Boat Racing.

£1.95 PLUS 4/C16+16K.

CHAMPIONSHIP TENNIS

A Tennis Simulation quite capable of creating the drama of the Centre Court. Available June.

C16/PLUS 4 £5.95

MICRO TEXT

Ideal for home use. Facilities include Centring and full Left and Right justification.

£1.95 (low), £10.95 (high).

MICRO BASE

An easy to use data base with full sort facilities. Ideal for keeping track of records or stamp collections. Click commands will find a very handy for finding names and addresses of members. Designed to print self-address labels. Member names and addresses can be entered for printing.

£1.95 (low), £10.95 (high).

C16 and PLUS 4 REFERENCE BOOK

The book has been specially written for ANCO. It is a comprehensive guide for a beginner and a professional programmer. June.

£7.95



ANCO

ANCO SOFTWARE, 4 WEST GATE HOUSE, SPITAL STREET, DARTFORD, KENT, DA1 2EH.
Telephone: 0323 62613/62616.

Mail Order: Payment by Cheque, P.O., Visa or Access. 24 Hour Credit Card Service: 0322 626635.

Choosing the right computer is a good start — but can you find the right software?



At SUPERSOFT we're very conscious of the fact that people who spend several hundred pounds on computer equipment are looking to do rather more than play Space Invaders.

Financial planning is a rather grand name for something you've been doing all your life — making ends meet! Perhaps if Mr. Microbe had used **BASICLC 3**, he would have been able to balance the books a little better.

For home, club or small business use **BASICLC 3** should pay for itself in no time at all; for larger companies we recommend **BASICLC 3**, one of the few really valuable programs that you can learn to use in a day.

Although your Commodore 64 is a powerful musical instrument you need to be a pretty good programmer to understand how it all works. Unless, of course, you buy **MUSIC MASTER**!

To use **MUSIC MASTER** requires no prior musical knowledge, though in the hands of an experienced musician it will prove an invaluable tool. You don't need to know the first thing about programming either! **MUSIC MASTER** is the musical equivalent of a word processor, remembering the notes you play and allowing you to replay and edit them as you wish.

INTERSECTOR PILOT is a space flight simulator. Nowadays simulations are widely used to train pilots and astronauts because — to be frank — it's a lot cheaper (and safer) than the real thing!

Imagine, if you will, life in the 23rd century: space travel is commonplace, and on the outskirts of the galaxy the first war between civilisations is being fought. A shortage of trained pilots has prompted the Federation to develop a computer simulation that allows new recruits to gain experience without paying for their mistakes with their lives. With the aid of your Commodore 64 you too can learn to pilot the *Intersector 643* mail. But be warned — this is no game!

Other SUPERSOFT products include the **MIKRO ASSEMBLER** cartridge, the only assembler that's ideal for beginners yet powerful enough for the professional (most of our computers use it). The **WORMS** cartridge adds dozens of commands to Basic including tactical aids and disk commands; or an disk there's **MASTER 64**, a really comprehensive package for the keen programmer.

Of course, we do also publish games programs, and with classics like **STIX**, **QUINN** and **RAMBLAGE** in our range we are one of the major leaders. But we most enjoy coming up with the sort of programs that are going to be in use for months and years, not hours and days — the sort of programs that make you glad that you bought a computer — and glad that you bought SUPERSOFT!

You won't find SUPERSOFT products on the shelves of your local supermarket. But most specialist shops stock titles from our extensive range (and are prepared to obtain other programs to order). However you can also buy direct by sending a cheque (pre-paid orders are good too), by calling at our office, or over the telephone using your **ACCESS card**.



SUPERSOFT, Winchester House, Canning Road,
Wimbledon, Surrey, Middlesex HA9 7SL
Telephone: 01-891 1855